

DOCUMENT RESUME

ED 279 344

IR 051 816

AUTHOR Kriz, Harry M.
TITLE Science Orders Systems and Operations Manual.
INSTITUTION Virginia Polytechnic Inst. and State Univ.,
Blacksburg. Univ. Libraries.
REPORT NO AS-SPIRES1
PUB DATE 15 Dec 86
NOTE 88p.
PUB TYPE Guides - General (050) -- Computer Programs (101)

EDRS PRICE MF01/PC04 Plus Postage.
DESCRIPTORS Academic Libraries; Computer Software; *Database
Management Systems; Higher Education; *Information
Retrieval; *Library Automation; Library Collection
Development; *Library Technical Processes; *Online
Systems
IDENTIFIERS Approval Plans; *SCIENCE ORDERS

ABSTRACT

This manual describes the implementation and operation of SCIENCE ORDERS, an online orders management system used by the Science and Technology Department of Newman Library at Virginia Polytechnic Institute and State University. Operational since January 1985, the system is implemented using the SPIRES database management system and is used to (1) provide orders management for the collection development staff, (2) monitor the performance of approval plans, and (3) provide current order data to teaching and research faculty through the campus local area network. A description of the system provides information about the database structure, the STPROTGS public protocol subfile, report protocols, database maintenance tasks, loading orders data, receiving books, new books list preparation, purging aged orders, and database management. Sixteen appendices describe procedures for creating and sharing the SCIENCE ORDERS database, and list system file definers, file definitions, and SPIRES database management system protocols.
(Author/KM)

* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *

18

ED279344

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION
CENTER (ERIC)

- ☒ This document has been reproduced as received from the person or organization originating it.
- ☐ Minor changes have been made to improve reproduction quality.
- Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

SCIENCE ORDERS Systems and Operations Manual

Document Number AS-SPIRES1

December 15, 1986

Harry M. Kriz

Published by
Automation Services Department
University Libraries
Virginia Tech

BEST COPY AVAILABLE

"PERMISSION TO REPRODUCE THIS
MATERIAL HAS BEEN GRANTED BY

Harry M. Kriz

TO THE EDUCATIONAL RESOURCES
INFORMATION CENTER (ERIC)."

IR 051816

ABSTRACT

This document describes the implementation and operation of SCIENCE ORDERS, an online orders management system used by the Science and Technology Department in Newman Library, Virginia Polytechnic Institute and State University. The system has been operational since January 1985.

The SCIENCE ORDERS system provides orders management for the collection development staff, monitors the performance of approval plans, and provides current order data to teaching and research faculty through the campus local area network. SCIENCE ORDERS is implemented using the SPIRES database management system running on an IBM mainframe under IBM's VM/CMS operating environment.

This manual provides all details concerning the operation of the database at Virginia Tech. In addition, it provides all details necessary for establishing the online database system at other locations. File definers, file definitions, and SPIRES protocols are listed in their entirety.

PREFACE

In October, 1984, the University Libraries at Virginia Polytechnic Institute and State University (Virginia Tech) instituted new computerized acquisitions procedures. Library Assistants in the subject reference departments, which were responsible for all book selection, took on the task of data entry for orders using terminals connected to the University's mainframe computer through the campus local area network. The new procedures allowed selection personnel to enter orders directly into the Library's batch mode Acquisitions system. Prior to this new procedure, orders typed on paper in the subject reference departments had to be keyed into the computer system by Acquisitions Department staff. The new procedures eliminated this duplicate keying. In addition, the subject reference departments gained access to all firm order data in machine readable form.

In January, 1985, the Science and Technology Department instituted the SCIENCE ORDERS database. This SPIRES based system captured the machine readable data created for the Acquisitions Department's system, thereby eliminating the need for maintaining the paper file of Science and Technology Department orders which had contained several thousand slips of paper.

The most important effects of the SCIENCE ORDERS database were in the areas of improved services to the library user community and increased efficiency in the Library's internal operations. For the first time the Library was able to publish a New Books List. The first New Books List was published in the first month of operation of the new database, and this has been a well received service ever since. Faculty were given direct access to orders information since they could access the database through terminals in their offices. The quality of the collection improved as Departmental staff took advantage of the increased efficiency of the online system to increase the breadth and depth of their collection development efforts. Staff in public services became thoroughly familiar with the operation and maintenance of an online database system. Insights gained from this experience were enormously helpful in gaining greater understanding and appreciation of the problems faced by technical services staff in managing the online public access catalog. Librarians who had been maintaining their personal orders files were able to save a great deal of labor by depending on the Department's shared database.

In the first year of operation of SCIENCE ORDERS, the system enabled the Science and Technology Department to double the output of orders from slightly more than 6000 orders per year to more than 13,000 per year. The increased efficiency allowed professional librarians and library assistants to devote greater effort to the development of the more difficult parts of the collections, including conference proceedings and society publications. The Acquisitions Department found that shared access to the online data was useful in solving problems with the approval plans and in making handling of the approvals more efficient. The double handling previously required for cost assignment was eliminated.

SCIENCE ORDERS was conceived as a system which could be operated and maintained by library reference department staff without the intervention of computer specialists. For these employees, this manual describes in detail the operation of the SCIENCE ORDERS database.

Details of the design of the SCIENCE ORDERS system, including the file definer, file definition, and protocols, are contained in the appendixes to this manual. There is sufficient detail to allow implementation of this system at other institutions where SPIRES is available. Also, the detailed listings are useful in gaining a complete understanding of the workings of SCIENCE ORDERS, especially if it becomes necessary or desirable to modify or customize the system in any way.

The author wishes to thank all those who contributed to the development of SCIENCE ORDERS during the last two years. Professor Victoria Kok provided initial impetus and continuing encouragement throughout the project. Professor Ludeman Eng of the College of Veterinary Medicine introduced us to the techniques of using SPIRES. Mr. William Ranck of the Virginia Tech Computing Center provided critical technical know how at several stages of development. Ms. Cathy Epperly was always willing to learn new procedures, and unlearn old procedures, in managing the database throughout its evolution. Ms. Anna Dickerson and Ms. Eleanor Garrison were diligent in putting the system through its paces on a daily basis.

For those who helped so much in developing SCIENCE ORDERS, and for those who use it every day, I hope that the project has made their jobs more interesting and their work more enjoyable.

TABLE OF CONTENTS

1.0	Introduction	1
1.1	Uses of Science Orders	1
1.2	Other Sources of Information	1
1.3	Searching SCIENCE ORDERS	2
1.3.1	Trying the system	2
1.3.2	Notes on searching	4
1.4	Outline of the Manual	4
2.0	Database Structure	5
2.1	Database Elements	5
2.1.1	System supplied elements	5
2.1.2	Required elements	5
2.1.3	Optional elements	6
2.2	Levels of Access	7
2.2.1	Owner privileges	7
2.2.2	Private privileges	8
2.2.3	Public access	8
2.2.4	Changing access privileges	8
3.0	The STPROTOS Subfile	9
4.0	Report Protocols	10
4.1	The ..printout Protocol	10
4.2	Miscellaneous Report Protocols	11
5.0	Database Maintenance Tasks	12
6.0	Loading Orders Data	13
6.1	Summary	13
6.2	Detailed Procedures	14
6.2.1	Part 1: Sending orders to the Acquisitions and Science Orders userids	14
6.2.2	Part 2: Loading SCIENCE ORDERS	14
6.2.3	Part 3: Error correction	15
6.2.4	Sample session for Part 1	16
6.2.5	Sample session for Part 2	16
6.2.6	Sample session for Part 3	17
6.3	NOTES on disk space	17
7.0	Receiving Books	19
7.1	Summary	19
7.2	Detailed Procedures	19
8.0	New Books List Preparation	22
8.1	Summary	22
8.2	Detailed Procedure	23
9.0	Purging Aged Orders	25
9.1	Firm Orders	25
9.1.1	Summary:	25

9.1.2 Detailed firm order procedure for December 1986:	26
9.2 Taylor-Carlisle Orders	29
9.2.1 Summary:	29
9.2.2 Detailed procedure:	29
10.0 Database Management	33
10.1 Disk Space	33
10.2 Backup Copies	33
10.3 Available Space Recovery	34
10.3.1 Subfile status	35
10.3.2 Available space recovery (AVSPREC)	35
10.3.3 Database reload	36
10.3.3.1 Background	36
10.3.3.2 Procedures summary	36
Appendix A. SCIENCE ORDERS File Definer	38
Appendix B. SCIENCE ORDERS File Definition	41
Appendix C. Creating the SCIENCE ORDERS database	48
C.1 Procedures Outline	48
C.2 Detailed Procedure	48
Appendix D. Shared Public Access	50
D.1 Establishing the 200 Disk	50
D.2 Sharing the Database	52
D.2.1 Notes	52
Appendix E. ORFLOAD PROTOCOL	53
Appendix F. ROUTE1 DATA File	57
Appendix G. NEWALL PROTOCOL	60
Appendix H. NEWSELCT PROTOCOL	62
Appendix I. ...PRINTOUT PROTOCOL	65
Appendix J. COST PROTOCOL	72
Appendix K. SEARCHER PROTOCOL	74
Appendix L. STPROTOS File Definer	75
Appendix M. STPROTOS File Definition	76
Appendix N. Public Entry Protocol	77
Appendix O. Private Entry Protocol	79
Appendix P. PRINTIPL SPISORT	80

1.0 INTRODUCTION

SCIENCE ORDERS is a SPIRES database which contains records for all books and serial titles currently on order in the fields of science and technology in the University Libraries at Virginia Tech. It is a public database, which means that anyone with a userid on the University's VMI mainframe computer system can search and view the records.

The system consists of the SCIENCE ORDERS database, together with several programs (protocols in the SPIRES terminology) which provide printouts of the most requested data in standard formats accessed through simple menu systems. Additional protocols are used to produce listings of new books and to capture machine readable data from the University Library's mainframe acquisitions system.

The SCIENCE ORDERS system is a user maintained system, meaning that it is owned and operated by the individuals who use the data. Computer specialists play no role in the processing of the data, nor in the production of data output, nor in the management of system resources.

1.1 USES OF SCIENCE ORDERS

SCIENCE ORDERS is used to:

- maintain records of titles on order in the fields of science and technology
- maximize the efficiency of the ordering process
- optimize the reliability of the ordering and receiving process
- monitor the cost of acquiring materials
- inform faculty and other library users about new books in Science and Technology
- monitor the performance of approval plans

1.2 OTHER SOURCES OF INFORMATION

This manual describes in detail the procedures for using and maintaining the SCIENCE ORDERS database. Also, it describes the technical details of creating the database and the protocols which implement its various functions.

This manual does not describe the detailed use of the Virginia Tech interactive computing environment, nor does it describe the features of SPIRES which are common to all SPIRES databases. Those not familiar with IBM's CMS and XEDIT should consult the appropriate manuals available from the Computing Center. A list of these documents is available through the DOCSLIST command in CMS. Copies of any manual can be printed with the DOCS command.

Those not familiar with SPIRES, should consult the following SPIRES documentation:

- SPIRES SEARCHING AND SORTING VM/CMS PRIMER
- GUIDE TO DATABASE DEVELOPMENT
- SPIRES SEARCHING AND UPDATING
- FILE DEFINER
- SPIRES PROTOCOLS

These documents are available using the following sequence of commands from CMS ready mode:

```
SPIRES
SPIRDOC
Follow the prompts
```

Computing Center document SP01 (Introduction to the SPIRES database management system) is a useful introduction to SPIRES.

1.3 SEARCHING SCIENCE ORDERS

The standard techniques for searching SCIENCE ORDERS are the same as those for searching any SPIRES database. These procedures are described fully and clearly in the documentation mentioned above. Those familiar with SPIRES can use the system immediately without instruction. Those unfamiliar with SPIRES may gain an initial understanding of the SCIENCE ORDERS system by trying the search described below.

1.3.1 Trying the system

Records in SCIENCE ORDERS are indexed in a variety of ways, including keywords in titles and individual words in author's names (both personal and corporate), among others. The following search uses these indexes. In the examples, text in brackets, e.g. [text], is to be supplied by the searcher. Do not type the brackets themselves.

```
logon [your userid]
SPIRES (to enter the SPIRES environment)
SELECT SCIENCE ORDERS (to select the desired database)
```

At this time an opening message describing the database is displayed.

Since every word in every title is indexed, you may search for a given title or set of titles simply by issuing the appropriate search command followed by any title words given in any order.

For example, to find all the titles now on order which contain the words PLANT SCIENCE, issue the command

FIND TITLE PLANT SCIENCE

(case of commands and title words is not regarded by the system. The search result would be the same
for : find title plant science
or even for : find title science plant)

To put the records in alphabetical order by title before listing them on the screen (this is unnecessary if the number of titles in the result is small), issue the command

SEQUENCE TITLE

To display the result on the screen, issue the command

TYPE

To find all the books by a particular author which are on order, issue the command

FIND AUTHOR [author's name, any words in any order]

At every step, SPIRES provides online help. Simply enter HELP in response to the SPIRES prompt.

To leave the system, issue the command

EXIT

which returns you to CMS ready mode. You may also issue the CMS logoff or log hold commands. In fact, all CMS commands are passed through when in a SPIRES session.

TIPS:

1. Narrow your search by specifying additional search terms if you find a large result for a search on a title word or words. You can do this by issuing a new FIND command with more search terms, though it is more efficient to issue the command:

AND [index name] [search term]

For example, to narrow a search for book titles with the word AGRICULTURE to those which also contain the word PLANTS proceed as follows:

FIND TITLE AGRICULTURE

Then at the SPIRES prompt:

AND {TITLE} PLANTS

Note that the index name TITLE is optional here as the search will be done on the index named in the preceding command unless a new index is named.

2. Issuing the TYPE command when the results of a search contain many entries will lead to a time-consuming listing. You can halt typing to the screen using the CMS **ht** command.

1.3.2 Notes on searching

- Some information in the database is not accessible in public mode. In particular, the requestor index cannot be searched by the public, and the names of those requesting a given title are hidden in public mode. Individuals wishing to have a list of titles they have requested may get a printout through the Science & Technology Department.
- It is a simple matter to obtain printouts of search results. Anyone wishing to have information on producing printouts should contact the Science & Technology Department by phone or through LocalNet at userid SCITECH AT VM1.

1.4 OUTLINE OF THE MANUAL

The chapters to follow describe the following:

- Database structure
- Summary of scheduled tasks
- Loading data into the database
- Receiving books
- Preparing the new books list
- Managing the database and associated disk files

The appendixes provide listings of the file definition and protocols, as well as the technical procedures necessary to create the SPIRES database within the Virginia Tech CMS environment.

2.0 DATABASE STRUCTURE

This chapter describes the elements or fields which make up each record in SCIENCE ORDERS. Some other features of the database are also described, including privilege levels for different access classes. The definitive description in the technical language of the file definer and file definition is provided in the Appendixes.

2.1 DATABASE ELEMENTS

SPIRES uses the term "element" as the name for a field in a record in the database. This section describes the elements in a SCIENCE ORDERS record. Note that the first three elements, namely the ID, DATE.ADDED, and DATE.UPDATED, are supplied automatically by the system. These elements occur in every record. The title element is a required element and must be entered for a record to exist. All other elements are optional and need not occur in a record. Unless otherwise noted, each element may occur no more than once in a record. Most elements have been assigned at least one alias, which is an alternative, usually shorter, form of the element name.

2.1.1 System supplied elements

ID is a unique identifier for each record assigned in serial order by the SPIRES system.

DATE.ADDED (alias da) is the date the record was added to the database. For practical purposes it is treated as the order date. The date is supplied automatically by the system. DATE.ADDED is indexed. For approval plans, see also the elements ANNOUNCED and AVAILABLE.

DATE.UPDATED (alias du) is a system supplied date which is updated every time the record is revised.

2.1.2 Required elements

A required element must occur at least once in each record in the database.

TITLE (alias ti, t) is required to occur once, and only once, in each record. It may contain any text characters. Each word in the title, except articles, conjunctions, and prepositions, is indexed in the TITLE index. The ISBN, ISSN, or Faxon Title Number is usually entered as part of the

title, as is the series added entry where appropriate. The title is entered according to the record found in OCLC or as supplied by the approval vendor for unpublished titles. Otherwise, the title as supplied by the publisher is entered.

2.1.3 Optional elements

Optional elements are not required to occur in each record.

AUTHOR (alias au, a) may contain any text characters describing authorship. Each word in the author element is indexed in the author index, with the exception of articles, conjunctions, and prepositions. Personal authors, corporate authors, and names of conferences are entered here.

PLACE of publication is entered here. Any text may be entered. PLACE is not indexed.

PUBLISHER (alias pu, pub) is entered here. Any text may be entered. This element is not indexed.

EDITION (alias ed) may contain any text describing the edition being ordered. This element is not indexed.

PUBDATE (alias pd, year) is the date of publication entered as text. It is not indexed. A range of dates may be entered.

VOLUMES (alias vols) may contain up to 3 characters entered as text to indicate the number of volumes ordered for the title. This element is not indexed.

LISTPRICE (alias price) contains any text denoting the price of the title. This element is not indexed. A text element is used rather than a numeric element because of the variety of entries used to indicate price.

COPIES contains an integer denoting the number of copies of the title to be ordered.

RUSHORD contains the numeral 1 to indicate a RUSH ORDER. Otherwise enter nothing. This element is not indexed.

RUSHCATA contains the numeral 1 if the item is to be RUSH CATALOGED. Otherwise enter nothing. This element is not indexed.

INSTRUCTIONS (alias specialinst) contains any text describing special instructions for processing the item after receipt, such as CATALOG REF/SCI, or SEE REQUESTOR BEFORE CATALOGING. This element is not displayed in the public access mode. It is not indexed.

REQUESTOR (alias req) contains any text to indicate the person requesting that the title be ordered. The proper form of entry is LIBRARIAN/OTHER REQUESTOR. The Librarian's name must come first to allow consistency in

subsequent processing. Each word in this element is indexed in the REQUESTOR index. This element is not displayed in the public access mode.

ACCOUNT (alias ac, acnt, fund, fd) may contain any text up to a maximum of 6 characters denoting the account to which the order is to be charged. The text will be capitalized by the system and indexed in the ACCOUNT index. Enter the standard departmental account acronyms.

CALLNO (alias call, c) contains the call number of the title as text. It will be capitalized by the system and indexed in the CALLNO index.

OTHER.ORDERS (alias statord) contains information about any other orders for this same title. This element is not indexed.

VENDOR (alias vend) is used primarily for the approval plans, most likely only the Taylor-Carlisle plan which announces titles in advance of publication. For firm orders, the vendor is not known to the Science & Technology Department. The element contains any text and each word is indexed in the VENDOR index. See also the ANNOUNCED AND AVAILABLE which relate to approval plans.

ANNOUNCED (alias date.announced) contains the announcement date from an approval slip. This element may occur no more than once. This date is not indexed.

AVAILABLE (alias date.due) contains the availability date (expected date of publication) from the approval slip. It must be entered in one of the valid SPIRES date formats. This element may occur no more than once. It is indexed in the AVAILABLE index.

RECEIVED contains the date the item is received. It must be entered in one of the valid SPIRES date formats. This element may occur no more than once. It is indexed in the RECEIVED index.

NOTES contains any additional information or notes to explain the status of the order. This element is not displayed in the public access mode, and it is not indexed.

2.2 LEVELS OF ACCESS

2.2.1 Owner privileges

The database is "owned" by userid SCIBASE. SCIBASE has full access privileges. It may enter, modify, and delete records, privileges not granted to any other class of user. All elements of the record are displayed to the owner.

2.2.2 Private privileges

Several userids are specifically authorized to view all elements of all records in the database. These userids may not add, modify, or delete records. Private privileges are assigned to the personal userids of each librarian collecting in the fields of science and technology, as well as to several userids used by the searchers for a variety of purposes. The Acquisitions Department has private access, as do the Library Director and the Assistant Director for Public Services and Collection Management. As of 11/86, the following userids had private privileges: SCIP, SCIS, SCISPI, SCIV, RENO, KOK, KRIZ, SCICO, REFER, TRAN, SCITECH, ACQPJM, RECSEC, RECSUP, GHERMAN, FOLTIN

2.2.3 Public access

SCIENCE ORDERS may be viewed by any userid on VM1. Userids accessing the database in this public mode may not enter, modify, or delete records. Userids accessing the database in the public mode are prevented from viewing those elements which would provide information about the requestor of the book. Thus, the REQUESTOR, INSTRUCTIONS, and NOTES elements are hidden in public access mode.

2.2.4 Changing access privileges

Access privileges are controlled through the subfile section of the file definition. They can be changed only by updating the file definition stored in the SPIRES subfile FILEDEF. The procedure is to select the FILEDEF subfile, transfer the SCIENCE ORDERS file definition to the active file, edit the definition to change the access privileges, and update the file definition in FILEDEF. Examination of the file definition in the appendix, and use of standard SPIRES record update procedures, will make the procedure clear.

3.0 THE STPROTOS SUBFILE

STPROTOS is a public protocol subfile containing three utility protocols which support SCIENCE ORDERS. Two of these protocols do no more than produce the opening messages displayed when SCIENCE ORDERS is selected. The third protocol is the ..printout command described elsewhere. The interaction of the STPROTOS subfile is transparent to the user of SCIENCE ORDERS.

Although STPROTOS is a public subfile, it can be selected only by the owner userid SCIBASE. This results from the CLEAR SELECT command included in the subfile section of the STPROTOS file definition. The owner userid may select the subfile and examine the protocols as records in the subfile. The subfile contains the following records:

- **ORDERSENTRYPRIV** which generates the opening message for those selecting SCIENCE ORDERS as a private subfile.
- **ORDERSENTRYPUB** which generates the opening message for those selecting SCIENCE ORDERS as a public subfile.
- **PRINTOUT** which generates the printed reports for the ..printout command.

The STPROTOS files reside on the 200 disk along with the SCIENCE ORDERS database. No maintenance is required for this subfile. Complete listings of the subfile definer, file definition, and protocol records are provided in the appendixes of this manual.

4.0 REPORT PROTOCOLS

Several protocols provide support for generating printed reports from SCIENCE ORDERS. In most cases these protocols conduct standardized searches, sometimes prompting the requestor for information needed as part of the search or as part of the final output. One protocol sets a format for a screen display found useful by the searchers. Sample print-outs should be generated by the reader to demonstrate the operation and output of the protocols. All protocols write to a file named OUTPUT FILE. Any existing file with this name will be over written.

4.1 THE ..PRINTOUT PROTOCOL

A single protocol, accessible through the command `..printout` generates several different types of standardized reports. The protocol is contained in the protocol subfile STPROTOS, a public SPIRES subfile which makes the protocol available to all authorized users.

The `..printout` command may be used by those userids which access SCIENCE ORDERS as a private subfile. These are the userids of individual bibliographers, several of the Science & Technology Department userids, Acquisitions Department userids, and userids of the Director and Assistant Directors who have requested such access. (To check a particular userid, enter the command `SHOW PRIVATE SUBFILES` at the SPIRES prompt. If SCIENCE ORDERS is listed as a private subfile, then the `..printout` command may be used.).

This command may also be invoked by userids accessing SCIENCE ORDERS as a public subfile. However, it will not run to completion because it incorporates the requestor element of the order record. A variation of the command could be provided for public users, but this would lead to increased system complexity. The availability of the `..printout` command is announced in the sign-on message for private users. It is not mentioned in the opening message to public users.

The following reports are generated automatically by `..printout`:

1. A standard format report for an arbitrary search result. The requestor must first have completed a search.
2. A standard format report for orders for a specific requestor. An option allows this report to be limited to RUSH orders only.
3. A standard format report for continuation orders for a specific requestor.

The protocol which generates these reports is listed in an appendix. To generate samples, use the following sequence of commands and respond to the prompts. Complete on-screen instructions are provided.

SPIRES
SELECT SCIENCE ORDERS
..PRINTOUT

4.2 MISCELLANEOUS REPORT PROTOCOLS

Several report protocols are executed from the active file on the userid which owns SCIENCE ORDERS. The protocol code is contained in a CMS file with the filetype PROTOCOL and a filename which is the name of the protocol. For simplicity, these protocols are executed through a CMS EXEC file which calls the protocol with a single command. The EXEC file has the same filename as the PROTOCOL file. The EXEC file is not necessary, but it provides the convenience of executing the protocol by a single command. If the EXEC were not used, the protocol would have to be invoked by first telling SPIRES to set the active file to the file containing the protocol. Listings of various EXECs are provided throughout this manual or in the appendixes as appropriate where the protocols themselves are listed.

Those protocols which generate the New Books list are described in other chapters and will not be discussed here.

COST PROTOCOL generates a price report arranged by requestor's name and subject account for all firm orders entered into SCIENCE ORDERS on a given date. It is invoked by issuing the command COST after selecting SCIENCE ORDERS. The files COST EXEC and COST PROTOCOL must be on the userid.

SEARCHER PROTOCOL sets a \$REPORT format found to be convenient by the searchers. The files SEARCHER PROTOCOL and SEARCHER EXEC must be on the userid. These files have been copied to several userids in the department.

PRINTIPL SPISORT was created to print a listing of all books on order. To print 6000 orders costs about \$20 in December, 1985. The printout was seldom if ever used, and the protocol has not been used since the end of 1985. A listing is provided in an appendix. If such a complete printout is desired, the protocol should be reviewed for correctness as some procedures may be out of date. Care should be taken that sufficient disk space exists to contain the OUTPUT FILE. The file PRINTIPL SPISORT must be on the userid. The procedure is invoked by issuing the commands:

USE PRINTIPL SPISORT
XEQ

5.0 DATABASE MAINTENANCE TASKS

Many tasks must be performed regularly to ensure the correctness, integrity, and security of the data in SCIENCE ORDERS. A single individual should be appointed data base manager to perform these tasks. This individual should be skilled in the use of XEDIT and SPIRES in the CMS environment. Only this individual should have access to the SCIBASE userid which owns the database.

1. Daily tasks

- Entering orders other than those from the weekly batch load. This includes approval plan orders which are batch loaded whenever there are approval slips to process.
- Updating records for titles received. This is done in batch mode as status notification forms (SNFs) for each cataloged title are received from cataloging.

2. Weekly tasks

- Capturing, batch mode processing, and batch loading into SPIRES the records created for the Library's batch mode Acquisitions System.
- Monitoring disk space.
- Purging old files from the owner userid.
- Maintaining adequate backup files.

3. Monthly tasks

- Purging older files on the userids used for data entry for the Acquisitions system.
- Preparing the New Books List and distributing it.
- Purging older records from the database.
- Monitoring problems with orders and reporting such problems to the requesting librarians.
- Backing up the database.
- Recovering available file space.

4. Semiannual tasks

- Monitoring lingering problems with aged orders.
- Reviewing procedures.
- Monitoring changes in the SPIRES system as they affect the operation of the database.

6.0 LOADING ORDERS DATA

There are two classes of data used to load orders records into SCIENCE ORDERS.

- Machine readable data from the Acquisitions Department's system. These represent firm orders for books and serials. This aspect of data entry is unique to Virginia Tech. The procedures capture machine readable data from disk files created for processing by the Library's batch mode Acquisitions system. Readers outside Virginia Tech may be interested in the method as it illustrates the use of the SPIRES protocol language to write a procedure which can capture any form of formatted data and prepare it for loading into a SPIRES database.
- Data which must be keyed, either by interactive data entry using SPIRES prompting mode, or by editing and subsequently batch loading of a CMS file. Taylor-Carlisle approval plan orders constitute the bulk of this category. SCIENCE ORDERS contains the only machine processable records in the Library for the approval plan orders.

This section deals only with the first category of data, i.e. files which are produced by the procedures described in THE ACQUISITIONS SYSTEM: Reference Department's Procedure by Bill Dougherty & Betty McClellan, Library Systems Office, September, 1984. The data for approval plans can be keyed using the SPIRES prompting procedures in \$PROMPT, or they can be placed in a CMS file using XEDIT and batch loaded using SPIBILD. Procedures for batch loading using the standard SPIRES input format are fully described in the SPIRES documentation and will not be further discussed here.

6.1 SUMMARY

1. The Departmental searchers verify all data and prepare a data entry form, typically a printout from OCLC.
2. The order data is keyed through an interactive prompting system into a specially formatted CMS file on the acquisitions userid for each librarian.
3. Once each week, the resulting CMS files are forwarded to the Acquisitions Department for further handling and placement of orders.
4. Copies of the CMS files also are sent to the userid which owns the SCIENCE ORDERS database.
5. The CMS files are processed by a SPIRES protocol into a form readable by SPIRES and batch loaded into SCIENCE ORDERS.

The files ORFLOAD EXEC and ORFLOAD PROTOCOL must be on the SCIBASE userid. The file ORFLOAD DATA is created by this protocol. Any existing file with this name is over written.

6.2 DETAILED PROCEDURES

6.2.1 Part 1: Sending orders to the Acquisitions and Science Orders userids

- Logon to the order userid for each librarian. The file containing the current week's orders is **ROUTE1 DATA**.
- Rename each **ROUTE1 DATA** to a name of the type: **IIImmdd DATA**

where III are the librarian's Initials, mm stands for the month, and dd stands for the day. Thus, when on the userid ACQHMK on March 12, the command is:

RENAME ROUTE1 DATA HMK0312 DATA

The command can be entered from **FLIST** by typing over the entry for **ROUTE1 DATA** using **REN / HMK0312**.

- Send the renamed file to the Acquisitions Department (userid **ACT2**) and to the userid which owns **SCIENCE ORDERS** (userid **SCIBASE**).

6.2.2 Part 2: Loading SCIENCE ORDERS

- Logon to **SCIBASE**.
- Receive the files sent from the Librarian's userids. These files are thought of as ORFs because they follow the old ORF (Order Recommendation Form) format.
- Use the **XEDIT PUT** command to put each librarian's ORFS file into a single file named **ORFSmmdd DATA**.
- Enter **SPIRES**.
- Issue the command **ORFLOAD**
Respond to the prompts.

The **ORFLOAD** protocol processes the ORF equivalent file into a new file named **ORFLOAD DATA** in which the orders are displayed in the standard **SPIRES** input format. During the processing, note the number of orders.

- After processing is complete, issue the following commands:

```

SPIBILD
USE ORFLOAD DATA
BATCH SCIENCE ORDERS

```

- Make sure all orders load.

6.2.3 Part 3: Error correction

The SPIBILD batch loader reports its progress for each record loaded. Should a record fail to load from the input file, the line number where the error occurred in the CMS file ORFLOAD DATA is printed on the screen. This makes it possible to correct the data and try a second time to load it.

In loading orders during the first six months of operation, the only errors were caused by semicolons (;) in the titles, by missing blank lines between orders in the original Acquisitions data file processed by ORFLOAD, and by the symbol Y being used instead of the numeral 1 to designate a rush order. Care must be taken that semicolons are not used in inputting titles. Checking the number of lines in the input file before it is processed by ORFLOAD can assure that there are no missing lines in a record, as the number of lines must be divisible by 14. Care should be taken that the proper characters are input when required. After the first few months, errors were no longer a problem as the data entry personnel became familiar with the system.

If errors do occur in loading SCIENCE ORDERS, the problem records can be corrected and loaded correctly through the following procedure. Extract the problem records from the file ORFLOAD DATA by noting the lines on which the errors occurred in the input file. XEDIT the input file ORFSmdd and use the PUT command to put the incorrect records into a new file called FIX MEMO. Edit the records into their correct form. (The filetype MEMO is used for convenience as XEDIT defaults to mixed case for this filetype). The file FIX MEMO is then used in place of ORFLOAD DATA in the procedure outlined in Part 2.

SPIRES provides a mechanism to automatically send lines of data which do not load to another file for further editing. This procedure has not been used because by the time it was discovered no errors were occurring. If error handling becomes a problem, this capability should be further explored.

6.2.4 Sample session for Part 1

Below is a sample session of sending orders for the week of February 5.

L ACQEG (log on to the userid for librarian EG)
Q (quit the order entry procedure which starts on entry)
FL (enter flist)

RENAME ROUTE1 DATA EG0205 DATA

Press the PF2 key (to refresh flist and see file is renamed)

SENDFILE EG0205 DATA to ACT2 (sendfile to Acquisitions)

SENDFILE EG0205 DATA SCIBASE (sendfile to Science Orders userid)

LOG HOLD

Make a note of which userid has been done to avoid duplication or
avoid missing a userid.

Repeat this sequence for all userids used for orders data entry.

6.2.5 Sample session for Part 2

Below is a sample session of loading the orders in Science Orders for the
week of February 5.

log on to SCIBASE

Receive the files of orders.

For each file

 XEDIT each file (in order to use the PUT command)

 PUT * ORFS0205 DATA (the first time this is issued a new file
 is created.)

 PF3 to quit out of XEDIT in each file and return to flist

After putting all files into ORFS0205 DATA, issue the command ORFLOAD

Respond to the prompts. Then note the number of orders.

The file ORFLOAD DATA should now be ready for loading into SCIENCE ORDERS.
Issue the following commands:

SPIBILD
USE ORFLOAD DATA
BATCH SCIENCE ORDERS

Note the number of requests, successes and failures.

Note the line numbers of any records which caused errors.

6.2.6 Sample session for Part 3

Correct the records which caused errors and load the corrected records into SCIENCE ORDERS. Records are corrected by putting them into a file called FIX MEMO, editing the file, and then batch loading the contents of FIX MEMO into SCIENCE ORDERS.

X ORFLOAD DATA

Go to lines where errors occurred in the BATCH SCIENCE ORDERS procedure and PUT the problem records into a file called FIX MEMO.

XEDIT FIX MEMO

Look for semicolons (;) in the titles, extra semicolons at the end of lines, or completely scrambled records. When records are completely scrambled the problem most likely occurs from a missing blank line between records in the original ORFSmmdd DATA file. It will be necessary to go to the original ORFS0205 DATA file and find the records that caused the error in processing. These records will have to be put into a new file to be processed by the ORFLOAD protocol. This requires some care and experience in file manipulation and editing. For this reason it is best to check for divisibility by 14 before processing the original ORFS file. Correction then is much simpler and less confusing.

FILE FIX MEMO

SPIBILD

USE FIX MEMO

BATCH SCIENCE ORDERS

6.3 NOTES ON DISK SPACE

Maintaining and operating the Science Orders database requires that one individual be responsible for the monitoring of disk space. The **q disk** command should be used to check available disk space. From time to time unused files, or files left behind by other processes, should be erased.

Processing and loading orders uses about 15% of the 1620 blocks of storage on the userid. Most of the disk space is taken up by the backup copies of the database itself.

Each week it is best to:

Make sure there is adequate room on the disk.

Erase old order files such as IIIImdd DATA, ORFSmmdd DATA, ORFLOAD DATA

Erase any lists generated by SPIRES protocols which are no longer needed including, IPL FILE, GENLIST FILE, OUTPUT FILE, etc.

Before sending orders it is imperative to verify that each order file has a total number of lines divisible by 14, since each order record occupies 14 lines. If there are too few lines, look for a missing blank line between records. If there are too many lines, it usually results from too many lines between records and there will be no problem. However, it is safest to correct the anomalies.

7.0 RECEIVING BOOKS

This section describes the procedures for setting the received status for a record in SCIENCE ORDERS.

After books are received and cataloged, a printed record of the new book is sent to the Science & Technology Department for distribution to the requesting librarian. The printed record is the Status Notification Form (SNF). Using the SNF, the SCIENCE ORDERS manager updates each order record in SCIENCE ORDERS with the information that the book has been received and the date on which the SNF was received. The latter date is taken as the date the book is received and available for use.

The received status on a record allows the record to be retrieved for inclusion in the New Books List, and it provides a method for purging the record from the database.

7.1 SUMMARY

1. For each SNF, find the title in SCIENCE ORDERS
2. Verify that the order record corresponds to the item received (some titles have more than 1 order)
3. Type the id, title, call number (usually blank) and received elements into the active file R MEMO
4. XEDIT R MEMO to enter the LC class, the date received, and the instruction to merge this new data into the existing record
5. batch process R MEMO to add the changes to the SCIENCE ORDERS database.

7.2 DETAILED PROCEDURES

All procedures must be carried out on the userid which owns the database as this is the only userid with update privileges. Several of the commands necessary to initiate and carry out these procedures have been incorporated into EXECS on the userid. The files R EXEC and ENDR EXEC must be on the userid. ORDERS EXEC is not required, but it has been installed on many userids to provide for a single command to call SPIRES and select SCIENCE ORDERS.

1. ORDERS EXEC contains the following commands:

```
&TRACE  
&STACK SELECT SCIENCE ORDERS  
EXEC SPIRES
```

2. R EXEC contains the following commands:

SET ACTIVE R MEMO
SET PF01 IN ACTIVE CON TYPE ID T RECEIVED CALLNO

3. ENDR EXEC contains the following commands:

&TRACE
CP SET PF01 FIN T

Listed below is a sample session:

Part 1: Placing received titles in the active file R MEMO

LOGON to SCIBASE

ORDERS (executes an exec which calls SPIRES, and
SELECTS SCIENCE ORDERS)

R (executes an exec which sets the active file to
R MEMO and sets the PF1 key to issue the command
which types the needed parts of the record into
the active file)

FIND TITLE [words from title]

(This may have to be repeated until
a single title forms the result)

TYPE (displays the final record on the screen for verification)

PF1 (places the command on the screen to type the necessary
elements into the active file)

<RETURN> (activates the command)

Part 2: Editing the data

After finding all titles and placing them in the active file, then
issue the commands:

ENDR (executes an exec to reset PF01)

X R MEMO (XEDIT the file with the received data.

this includes:

global change from ID = #; to MERGE #;
enter the LC class in the CALLNO element
enter the date received in RECEIVED
delete the record separator ***)

Part 3: Loading the received data

After editing all the received information, issue the commands

FILE

SPIBILD

USE R MEMO (to set the active input file to R MEMO)

BATCH SCIENCE ORDERS (processes the contents of the active
file into the database SCIENCE ORDERS).

After the completion of this procedure, the SNF's may be distributed to the librarians or other requestors. All the received data is in SCIENCE ORDERS.

8.0 NEW BOOKS LIST PREPARATION

8.1 SUMMARY

Prepare a listing of new books at the end of each month. Each librarian receives a list of all books received during the month. Each faculty library representative receives a list of selected new books. A copy of the selected list is posted in the Library.

Items excluded from the selected list are replacement copies, added copies, older materials, serial subscriptions, and standing orders. Following production of the lists, the received titles are purged from the database, and various database management procedures are carried out. The latter include file backup and disk space recovery.

The new books lists are produced by executing protocols contained in CMS files on the SCIBASE userid. The protocols are invoked by CMS EXEC files. For these protocols to work, SCIENCE ORDERS must be the selected database.

The files NEWALL EXEC, NEWALL PROTOCOL, NEWSELCT EXEC, and NEWSELCT PROTOCOL must be on the SCIBASE userid, in addition to EXEC and PROTOCOL files mentioned in previous sections of this manual.

1. The file NEWALL EXEC executes the NEWALL PROTOCOL. It contains the commands:

```
&TRACE
&STACK SET ACTIVE NEWALL PROTOCOL
&STACK XEQ
```

2. The NEWALL PROTOCOL is listed in an appendix. The protocol prompts for month and year, finds all books received during that month, and prints 6 copies in landscape output using the general list format.
3. The file NEWSELCT EXEC executes the NEWSELCT PROTOCOL. It contains the commands

```
&TRACE
&STACK SET ACTIVE NEWSELCT PROTOCOL
&STACK XEQ
```

4. The NEWSELCT PROTOCOL is executed only after the titles not wanted on the selected list have been removed from the database. NEWSELCT prompts for month and year, finds all books in the database received during that month, prompts for number of copies, and prints the selected number of copies in portrait output.

8.2 DETAILED PROCEDURE

1. Issue the command **ORDERS**
2. Issue the command **NEWALL**
3. Distribute a copy of the resulting printout to each librarian.
4. Edit the database to remove all received titles which are not wanted for inclusion on the Selected List of New Books:
 - a. Review the printout to determine which titles are not wanted on the public distribution list.
 - To find titles to eliminate on the basis of publication date, scan the year column looking for old titles. Circle the id number for removal. (Practice has been to remove virtually all titles published more than 2 years previously. Thus, in 1986, titles published in 1984 usually were not included on the Selected List.)
 - To find other titles not wanted on the selected list, scan the titles themselves looking for duplicates, subscriptions, and non-science titles (for example, manuals for librarians in the Z classes). Circle the ids for these titles also.
 - b. Remove from the database the records for the circled IDs.
5. Issue the command **NEWSELCT**
Select the option to print only 1 copy when you are prompted to do so.
6. **NEWSELCT** puts a copy of the selected list into the CMS file **NEWBOOKS FILE**. This disk file is edited to print the final version of the Selected List of New Books:
 - a. After checking that all is in order on the copy produced by **NEWSELCT**,
XEDIT a new CMS file **NEWBOOKS SCRIPT**
GET the file **NEWBOOKS FILE**
Add headings to each LC class of call numbers
 - b. When editing is finished, print a sample copy by issuing the following commands:
SETP 3800P10 co 1
PRINT NEWBOOKS SCRIPT (CC
7. If the sample is acceptable, print 41 copies of the final version of the selected list by issuing:
setp 3800P10 co 41
print NEWBOOKS SCRIPT (cc

8. Distribute the copies to the library representatives and post 1 copy in the Department.

9. Remove from the database all records for the titles received during the month covered by the Selected New Books List. Use the following sequence of commands:

```
FIND RECEIVED [month year] where month and year represent the  
month and year of receipt of the titles to be removed.  
FOR RESULT  
REMOVE ALL  
ENDFOR
```

10. Perform database management tasks. See the database management section of this manual.

9.0 PURGING AGED ORDERS

Normal receiving procedures cannot clear all orders from the SCIENCE ORDERS database. There must be a periodic, systematic purge of the orders file in order to maintain the accuracy and currency of the records. Clearing should be done soon enough following placement of an order to catch problems, but not so soon that an excessive amount of checking must be done to track what are only the normal time lapses in various stages of the delivery and cataloging process.

Titles are not cleared from the database as part of the routine receiving process for several reasons including:

- Some titles are dropped by Taylor-Carlisle because information becomes available which indicates the previously announced book does not match the Library's profile. (A procedure for catching these titles was begun at the beginning of 1986).
- Some titles are never published.
- Some titles are changed prior to publication, and the title as received cannot be matched with the order record.
- Sometimes the Department does not get an SNF from Cataloging, or is not notified that an order has been canceled for any number of reasons.

The following procedures for systematically purging aged orders were developed over the 12 month period beginning in November, 1985. Firm orders and Taylor-Carlisle orders must be treated separately. Firm orders are purged depending on the date the record was added to the database. Taylor-Carlisle approval orders are purged based on the availability date provided by the vendor.

9.1 FIRM ORDERS

9.1.1 Summary:

Make an initial search for firm orders 12 months following the placement of the order. Search VTLS first. If the title is found in VTLS and not due for inclusion in the next New Books List, remove it from SCIENCE ORDERS. Then search the Acquisitions Department's In Process List (IPL). Titles which are not in VTLS and not in the IPL should be printed out for review by the requesting librarian.

Make a second search 18 months following placement of the order. Items not yet in VTLS and not on the IPL should be printed out for review by the requesting librarian. They should then be deleted from SCIENCE ORDERS. Titles which are still in the IPL because they are on order or sent to prep should be left in SCIENCE ORDERS, with an annotation placed in

the NOTES field to indicate the status of the order and the date the note was added. Periodically, perhaps once per year, the titles which have been left in the database following the 18 month check should be checked again so that a final disposition of the record can be made.

9.1.2 Detailed firm order procedure for December 1986:

1. **Part 1: Prepare printouts of aged firm orders.** After the New Books List is printed, and after the previous month's received books are removed from SCIENCE ORDERS, prepare printouts of old orders:

- a. Search for the firm order titles added to SCIENCE ORDERS 12 months earlier:

```
EXAMPLE FOR 12/1/86:  find da 11/85
                        and not vendor Taylor-Carlisle
                        for result where received occurs = 0
                        stack all
                        endfor
                        ..printout
```

- b. Respond to the ..printout prompts as follows:

- 1) Select the general list option (No. 1 on the menu)
- 2) Answer Y (Yes) when the routine asks if you want an alphabetical list.
- 3) When prompted, enter the subtitle: **Firm orders from 11/85**
- 4) Answer Q (Quit) when the routine asks if you want to quit the printout routine.

- c. Search for firm order titles added to SCIENCE ORDERS 18 months earlier:

```
EXAMPLE FOR 12/1/86:  find da 5/85
                        and not vendor Taylor-Carlisle
                        for result where received occurs = 0
                        stack all
                        endfor
                        ..printout
```

- d. Respond to the ..printout prompts as follows:

- 1) Select the general list option (No. 1 on the menu)
- 2) Answer Y (Yes) when the routine asks if you want an alphabetical list.
- 3) When prompted, enter the subtitle: **Firm orders from 5/85**
- 4) Answer Q (Quit) when the routine asks if you want to quit the printout routine.

2. **Part 2: Search the titles on the 12 month and 18 month firm order printouts:**
 - a. Search the titles in VTLS.
If a title on the list is found in VTLS, circle it on the list.
Remove the circled titles from SCIENCE ORDERS.
 - b. Search the remaining titles in the IPL.
Circle those titles which are **NOT** found in the IPL.
Annotate those titles which **ARE** found in the IPL, noting if a title has been sent to cataloging or been reordered.
3. **Part 3: Prepare printouts for 12 month old unreceived firm orders.**
After the 12 month printout from Part 1 is fully annotated as described in Part 2, use the list to do the following:
 - a. Annotate the SCIENCE ORDERS record for each title which was found in the IPL indicating new order dates or in process status. Enter the annotations in the NOTES field using the SPIRES MERGE command. If the IPL contains a new order date, enter the new date in the DATE.ADDED element so the record will be searched for purging under the new order date. In such cases, it may be easier to update the record by using TRANSFER to place a copy of the record in the active file for editing. The original order date can be entered in the NOTES field, along with the date of the IPL which is used to update the record. On the printout, cross out those titles which have been annotated as they are of no further interest in the 12 month review.
 - b. Use the SPIRES stack command (STACK [id number]) to place each title **NOT** found on the IPL into the stack. Remember to clear the stack before starting.
 - c. Issue the following commands:
SEQ REQ T
..PRINTOUT
 - d. Respond to the ..printout prompts as follows:
 - 1) Select the general list option (No. 1 on the menu).
 - 2) Answer N (NO) when the routine asks if you want an alphabetical list
 - 3) When prompted, enter the subtitle:
Firm orders placed 11/85, not in VTLS, not in IPL, will keep in SCIENCE ORDERS until 5/87
 - e. Circulate the resulting list to the librarians for any action they may wish to take in checking on the materials.
 - f. To produce additional copies of the list, immediately after quitting the ..printout command issue the following commands:
SETP 3800L10 CO 6 [or any other number of copies]
PRINT OUTPUT FILE (CC
SETP RESTORE

(The output of the ..printout command is written to a file named OUTPUT FILE. The commands above will print 6 copies of that file using the correct font size and line spacing. Then the printer will be restored to its original settings.)

4. **Part 4: Annotate or remove 18 month old firm order records from SCIENCE ORDERS.** After the 18 month printout from Part 1 is fully annotated as described in Part 2, use the list to do the following:
 - a. Annotate the SCIENCE ORDERS record for each title which was found in the IPL, using the NOTES field to record the current status of the order. If the IPL contains a new order date, enter the new date in the DATE.ADDED element so the record will be searched under the new order date. The original order date can be entered in the NOTES field, along with the new order date, and the date of the IPL which is used to update the record.
 - b. Use the SPIRES stack command (STACK [id number]) to place each title NOI found on the IPL into a search result. Remember to clear the the stack before starting.
 - c. Issue the following commands:
SEQ REQ T
..PRINTOUT
 - d. When in the PRINTOUT routine, do the following
 - 1) Select the general list option (No. 1 on the menu).
 - 2) Answer N (NO) when the routine asks if you want an alphabetical list
 - 3) When prompted, enter the subtitle:
Firm orders placed 5/85 not in VTLS, not in IPL, removed from SCIENCE ORDERS
 - e. Circulate the resulting list to the librarians for any action they may wish to take in checking on the titles.
 - f. To produce additional copies of the list, immediately after quitting the ..printout command, issue the following commands:
SETP 3800L10 CO 6 [or any other number of copies]
PRINT OUTPUT FILE (CC
SETP RESTORE
 - g. Remove the titles not found in the IPL from SCIENCE ORDERS by issuing the following commands:
FOR STACK
REMOVE ALL
5. **Part 5: Review lingering aged orders.** Every few months, orders which linger in the database beyond the 18 month cutoff because they are still in the IPL at the time of the 18 month review should be reviewed again. There will be very few. They should be brought to the attention of the requesting librarian, who may refer them to the Acquisitions Department.

9.2 TAYLOR-CARLISLE ORDERS

9.2.1 Summary:

Make an initial search for Taylor-Carlisle approval plan orders 12 months following the AVAILABILITY date. Search VTLS first. If the title is found (and not due for inclusion in the next New Books List), remove it from SCIENCE ORDERS. Then search the Taylor-Carlisle Open Order List. Titles which are not in VTLS and not on the Open Order List should be printed out and referred to the requesting librarian.

Make a second search for Taylor-Carlisle orders 18 months following the AVAILABILITY date. Items not yet in VTLS and not on the Taylor-Carlisle Open Order List should be printed out and referred to the requesting librarian. They should then be removed from SCIENCE ORDERS.

9.2.2 Detailed procedure:

1. **Part 1: Prepare printouts of aged Taylor-Carlisle orders.** After the New Books List is printed, and after the past month's received books are removed from SCIENCE ORDERS, prepare printouts of old orders:

- a. Search for the Taylor-Carlisle titles AVAILABLE 12 months earlier:

```
EXAMPLE for 12/1/86:  find vendor Taylor-Carlisle
                      and available 11/85
                      for result where received occurs = 0
                      stack all
                      endfor
                      ..printout
```

- b. Respond to the ..printout prompts as follows:

- 1) Select the general list option (No. 1 on the menu)
- 2) Answer Y (Yes) when the routine asks if you want an alphabetical list.
- 3) When prompted, enter the subtitle: TC orders available 11/85
- 4) Answer Q (Quit) when the routine asks if you want to quit the printout routine.

- c. Search for Taylor-Carlisle titles AVAILABLE 18 months earlier:

```
EXAMPLE FOR 12/1/86:  find vendor Taylor-Carlisle
                      and available 5/85
                      for result where received occurs = 0
                      stack all
                      endfor
```

..printout

- d. Respond to the ..printout prompts as follows:
- 1) Select the general list option (No. 1 on the menu)
 - 2) Answer Y (Yes) when the routine asks if you want an alphabetical list.
 - 3) When prompted, enter the subtitle: **TC orders available 5/85**
 - 4) Answer Q when the routine asks if you want to quit the printout routine.
2. **Part 2: Search the titles on the 12 month and 18 month Taylor-Carlisle printouts.**
- a. Search the titles in VTLS.
If a title on the list is found in VTLS, circle it on the list.
Remove the circled titles from SCIENCE ORDERS.
 - b. Search the remaining titles in the Taylor-Carlisle Open Orders List.
Circle those titles **NOT** found on the Taylor-Carlisle list.
Annotate those titles which **ARE** found on the Open Orders List indicating new delivery dates or changes in titles.
3. **Part 3: Prepare printouts for 12 month old Taylor-Carlisle orders.**
After the 12 month printout from Part 1 is fully annotated as described in Part 2, use the list to do the following:
- a. Edit the SCIENCE ORDERS record for each title which was found on the Open Orders List, entering the changed titles and new availability dates.
 - b. Use the SPIRES stack command (STACK [id number]) to place each title **NOT** found on the Taylor-Carlisle list into a search result. Remember to clear the stack first.
 - c. Issue the following commands:
SEQ REQ T
..PRINTOUT
 - d. When in the PRINTOUT routine, do the following:
 - 1) Select the general list option (No. 1 on the menu)
 - 2) Answer N (NO) when the routine asks if you want an alphabetical list
 - 3) When prompted, enter the subtitle:
TC orders AVAILABLE 11/85, not in VTLS, not in TC open order list, keep in SCIENCE ORDERS until 5/87
 - e. Circulate the resulting list to the librarians for any action they may wish to take in checking on the materials.
 - f. To produce additional copies of the list, immediately after quitting the ..printout command, issue the following commands:

SETP 3800L10 CO 6 [or any other number of copies]
PRINT OUTPUT FILE (CC
SETP RESTORE

4. **Part 4: Annotate or remove Taylor-Carlisle orders available 18 months previously.** After the 18 month list from Part 1 is fully annotated, use the list to do the following:
 - a. Annotate the SCIENCE ORDERS record for each title which was found on the Open Orders List, indicating the changed titles and new availability dates.
 - b. Use the SPIRES stack command (STACK [id number]) to place each title not found on the Taylor-Carlisle list into a search result. Remember to clear the stack before starting.
 - c. Issue the following commands:
SEQ REQ T
..PRINTOUT
 - d. When in the PRINTOUT routine, do the following:
 - 1) Select the general list option (No. 1 on the menu)
 - 2) Answer N (NO) when the routine asks if you want an alphabetical list
 - 3) When prompted, enter the subtitle:
TC orders AVAILABLE 5/85, not in VTLS, not in TC open order list, removed from SCIENCE ORDERS
 - e. Circulate the resulting list to the librarians for any action they may wish to take in checking on the titles.
 - f. To produce additional copies of the list, immediately after quitting the ..printout command, issue the following commands:
SETP 3800L10 CO 6 [or any other number of copies]
PRINT OUTPUT FILE (CC
SETP RESTORE
 - g. Remove the titles not found on the Open Orders List from SCIENCE ORDERS by issuing the following commands:
FOR STACK
REMOVE ALL
5. **Part 5: Review lingering old orders.** Titles still on the Taylor-Carlisle Open Orders List 18 months after the availability date should be treated as follow:
 - a. Search the title on OCLC.
 - b. If the title has been published and is owned by many other libraries, cancel the Taylor-Carlisle order and place a RUSH FIRM ORDER.

- c. If the title has not been published (which should be apparent if few if any libraries own the title, cancel the Taylor-Carlisle order.
- d. In both cases, a printout should be made for the requesting librarian and the records should be removed from SCIENCE ORDERS.

10.0 DATABASE MANAGEMENT

SPIRES databases are user maintained. This implies that the owner userid is responsible for technical aspects of database management, including backing up files, assuring that adequate disk space is available, and recovering unused file space in the SPIRES files. This is in contrast to database systems which are maintained by centralized support units. The most familiar example of a centrally maintained database system in the Library is VTLS, where those who enter and maintain the data are not responsible for the physical media on which the data is stored.

10.1 DISK SPACE

There needs to be adequate disk space for the database files on the 200 disk and on the A disk. This can be checked with the `q disk` command familiar to anyone maintaining CMS files. More disk space should be obtained if the files occupy more than 75% - 80% of the disk. Before requesting additional space, it should be verified that the RES file of SCIENCE ORDERS is using space efficiently. See the section below on available space recovery.

10.2 BACKUP COPIES

Practice for the first 2 years of using SCIENCE ORDERS was to have the working copy of the database on the 200 disk for shared access, and a backup copy of each of the files on the A disk. In addition, all ROUTE1 DATA files, all batch load files for the Taylor-Carlisle approval plan, and all batch received files were kept for at least 1 month so that the current version of the database could be recovered by taking the previous month's version and loading the batch files for the current month.

This procedure does not allow for recovering minor changes in records which might be entered during the month. In particular, changes which are made to individual records in the purging procedures would not be recoverable by these methods. This was judged to be of little or no consequence as the number of these changes in any given month is small. More complete backup procedures did not seem necessary given the functions of the database.

Should the SCIENCE ORDERS database take on an increased role, then it would be advisable to adopt more stringent backup procedures. These would involve weekly backups of the database, combined with a method of recovering all changes to the database during a given week. Tape backups might also be used. In adopting more stringent standards for backing up, it should be remembered that the Computing Center performs a tape backup of all disk drives on a weekly basis.

To maintain existing procedures:

1. At the end of each month, following production of the new books list and removal of the received titles, copy all files from the 200 disk to the A disk. This is most easily done from the filelist of the 200 disk. In the 1986 release of SPIRES the 200 disk is given a filemode Q when SCIENCE ORDERS is selected.

NOTE: It should be remembered that the COPY command first makes a copy on the A disk under a utility name, compares it to the original file, and only then erases an old file of the same name on the target disk. The result is that during part of the backup procedure there will be two copies of the RES file on the target disk at one time. If this file becomes very large, there may not be sufficient space on the target disk for the copy procedure unless the RES file from the previous backup procedure is erased prior to issuing the copy command.

2. Retain all of the renamed ROUTE1 DATA files on the individual acquisitions userids for at least 1 month so that data could be reloaded if necessary.
3. Retain all of the batch load files for the Taylor-Carlisle plan for at least 1 month so the data could be reloaded if necessary. Give these files appropriate names under which they can be saved after they are loaded the first time.
4. Retain all batch load files for the received titles for at least 1 month so the data could be reprocessed if necessary. Give these files appropriate names under which they can be saved after they are loaded the first time.

10.3 AVAILABLE SPACE RECOVERY

SPIRES subfiles with a many removes and modifications will accumulate unused storage space in the residual data set in CMS file FILENAME RES. This is the dataset containing the actual data. This is a particular problem with the SCIENCE ORDERS database since each month as many records are removed as are added. Space which had been allotted to the removed records is not automatically recovered by SPIRES. As a result, the ORDERS RES file can only grow in size, even though the number of records in the database remains constant, or even declines. SPIRES does not provide a mechanism for compacting the datasets once records are removed. It only provides a mechanism for assuring that unused space will be used by new records in the future. For a discussion, see the SPIRES manual FILE DEFINITION.

Proper file management, and control of the size of ORDERS RES, requires the following procedures to be run periodically. Otherwise, ORDERS RES will continue to grow even while the number of orders remains constant.

10.3.1 Subfile status

To analyze the use of file space, from SPIRES issue the following:

CALL SPIUTIL In SPIUTIL, subfiles may be selected, records searched and displayed, but no additions or modifications can be made.

SELECT SCIENCE ORDERS

Determine the status of the subfile by issuing

IN ACTIVE STATUS ALL

The result of these commands will be sent to the active file on the A disk, from where it can be printed.

Of particular interest is the figure for SPACE UTILIZED in ORDERS RES. If this figure drops much below 90%, the Available Space Chain should be examined to see if an AVSPREC should be run. For SCIENCE ORDERS, it should be. See section B.10.1 of the FILE DEFINITION manual for details.

When STATUS was run for SCIENCE ORDERS in February, 1986, after 13 months of activity, only 43% of the space in RES was utilized. At that date, nearly 14000 records had been loaded into the subfile but enough had been removed so that only about 6300 records remained. The RES file had gotten very large, with no increase in the number of records.

10.3.2 Available space recovery (AVSPREC)

The following procedure was used to run AVSPREC, which requires that the owner userid have w/r access to the disk containing the RES file, although there seems to be no change in the disk files, at least the date stamp remains unchanged. (SCIENCE ORDERS is on the 200 disk of its owner userid).

```
LDISK * 200 200 WR
SPIRES
CALL SPIUTIL
SELECT SCIENCE ORDERS
AVSPREC
LOG HOLD
```

AVSPREC writes to the ACTIVE FILE on the A disk. The program provides a listing of the available space chain. There was no apparent change in the database files on the 200 disk.

After running this in February, 1986, loading 144 records caused no increase in the size of the database's CMS files. ORDERS RES increased only

slightly in size during subsequent months when AVSPREC was run almost monthly.

10.3.3 Database reload

10.3.3.1 Background

A database reload of SCIENCE ORDERS was executed on 10/22/86 in order to compact the data files. The RES file utilization had decreased to about 40% at this time. Apparently AVSPREC was working to keep the file fairly constant in size, but the low utilization percentage was the result of operating the system during its first year without awareness of the available space problem or the AVSPREC routine. This information on reloading the database is included should the RES file continue to grow in the future. It is not expected that this could become a problem for many months. Monitoring the size of the files as part of disk space management will indicate the need for such a routine.

Should a reload become necessary, it would be best to review these procedures with a SPIRES consultant before proceeding.

10.3.3.2 Procedures summary

1. Record number of titles in the database as well as the next slot number so that running totals of activity over the years can be kept.
2. Process the subfile in SPIBILD to complete changes pending in the deferred queue.
3. Print the filelist for the Q disk.
4. Prepare backups of the database files using the CMS ARCHIVE utility. First copy the files from the Q disk to the A disk. Then use ARCHIVE to copy the files to tape.
5. Write a copy of the file definition from FILEDEF to the A disk. This should not be necessary as the definition in FILEDEF should not be destroyed when the database is zapped. However, it is a wise precaution to make the backup.
6. Erase the database copies on the A disk to provide working space.
7. Copy all records out of the database to a file named RELOAD DATA A. At 11:00 P.M., copying out the data took 25 seconds to write 30380 lines of data to 507 blocks with 5859 titles in the database.

- clear formats
 - set all elements desired, which should be every element except the ID. A new ID will be assigned when the records are reloaded. The DATE.ADDED (which is used as the original order date) will be entered as it was in the original file. The DATE.UPDATED will be modified to the date of the reload. It is easiest to set all elements except the ID by listing the elements to a file, building a command from this list, and executing the file as a protocol. This is because the ID (a slot key element) cannot be set by simply subtracting it from ALL elements. All elements except the ID must be listed explicitly.
 - set recsep 'add;'
 - display some records in the active file to check the display format
 - set active RELOAD DATA
 - for subfile
 - in active scan
 - endfor
8. ZAP FILE SCIBASE: SCIENCE. ORDERS
 9. Compile SCIENCE ORDERS
 10. Load the newly compiled database using the following procedures:

```

SPIBILD
CP SPOOL CONS START TO *
USE RELOAD DATA
BATCH SCIENCE ORDERS

```

The CP SPOOL command will send all messages to the reader for checking following completion of the reload. It is possible to disconnect from the system while the batch load is progressing. Use the command:

```
#cp disc
```

It is possible to logon at a later time using normal logon procedures. Then type B (for begin). The batch procedure may still be executing in SPIBILD and you will be able to read the messages. If the process has completed, issue the command:

```
SPOOL CONS STOP CLOSE TO *
```

APPENDIX A. SCIENCE ORDERS FILE DEFINER

The original file definition for SCIENCE ORDERS was generated by the SPIRES FILE DEFINER system using the file definer listed here. The resulting file definition is substantially that shown in the next appendix. The following changes were made directly to the file definition after generation by the file definer system:

1. Constraints and priv-tags were added directly to the file definition as the file definer did not produce these as documented in 1985.
2. Changes in access privileges have been made by editing the file definition. The file definition has been commented to indicate the nature and date of these changes.

```
FILE /AUTHOR H. M. KRIZ, Sci-Tech Department, Newman Library, 6354
SUBFILE SCIENCE ORDERS
CMD Show select
CMD Show subfile size
CMD Set format $prompt
  EXPLAIN This database lists titles on order or scheduled for
  EXPLAIN delivery through approval plans to the Science & Technology
  EXPLAIN Department, Newman Library. The file is purged periodically,
  EXPLAIN usually after printing of the monthly list of Selected New
  EXPLAIN books. No attempt is made to track the progress of an order
  EXPLAIN through the Library's Acquisitions and Cataloging Departments.
  EXPLAIN Please check VTLS or the Library In-Process List for the most
  EXPLAIN recent information about an order, or ask a Reference
  EXPLAIN librarian for assistance.
SUBFILE SCIENCE ORDERS/ ACCOUNTS SCIP, SCIS, SCIV, RENO, KOK, KRIZ, +
  SCICO, REFER, CRISSING, TRAN, SCITECH/ SWITCHES 3
CMD Show select
CMD Show subfile size
CMD Set format $prompt
SUBFILE SCIENCE ORDERS/ ACCOUNTS PUBLIC/ SWITCHES 3/ CONSTRAINT 1/+
  NOUPDATE 1/ NOS 1
CMD Show select
CMD Show subfile size
CMD Set format $prompt
GOAL/ Result Title
FIXED
KEY ID/ SLOT
ELEMENT DATE.ADDED, DA/ DATE ADD/ INDEX
  EXPLAIN This system supplied date can be taken as the date the order
  EXPLAIN was forwarded to Acquisitions. For approval plans, see the
  EXPLAIN elements ANNOUNCED and AVAILABLE for more information.
ELEMENT Date.updated, du/ Date update
  EXPLAIN This system supplied date tells the last time a record was
```

EXPLAIN revised.

REQUIRED

ELEMENT Title, ti, t/ text/ squ/ single/ index/ word/ exclude

EXPLAIN Enter the ISBN number at the end of the title.

EXPLAIN Enter the title as determined from the OCLC database, or in

EXPLAIN the case of approval plans, as supplied by the vendor. Other-

EXPLAIN wise enter the title as supplied by the publisher.

OPTIONAL

ELEMENT Author, au, a/ text/ squ/ single/ index/ word/ exclude

EXPLAIN This includes personal authors, corporate bodies, and

EXPLAIN names of conferences.

ELEMENT Place/ text/ squ/ single

ELEMENT Publisher, pu, pub/ text/ squ/ single

ELEMENT Edition, ed/ text/ squ/ single

ELEMENT Pubdate, pd, year/ text/ squ/ single

EXPLAIN Any text may be entered for date ranges, etc.

ELEMENT Volumes, vols/ text/ length 3/ single

ELEMENT Listprice, price/ text/ single

ELEMENT Copies/ integer/ single

ELEMENT Rushord/ integer/ single

EXPLAIN Enter the numeral 1 for a RUSH ORDER. Otherwise enter

EXPLAIN nothing.

ELEMENT Rushcata/ integer/ single

EXPLAIN Enter the numeral 1 if the item is to be RUSH CATALOGED.

EXPLAIN Otherwise enter nothing.

ELEMENT Instructions, specialinst/ text/ squ/ single/ priv-tag 1

EXPLAIN Enter here any special instructions for processing the

EXPLAIN item after receipt, such as CATALOG REF/SCI, or SEE REQUESTOR

EXPLAIN before cataloging.

ELEMENT Requestor, req/ text/ squ/ single/ index/ word/ priv-tag 1

EXPLAIN Enter as LIBRARIAN/OTHER REQUESTOR. The Librarian's name

EXPLAIN must come first to allow consistency in subsequent processing.

EXPLAIN The '/' punctuation is required as the separator between

EXPLAIN the librarian's name and any other names.

ELEMENT Account, ac, acnt, fund, fd/ text/ single/ capitalize/ index/ length 6

EXPLAIN Enter the standard departmental account acronyms.

ELEMENT Callno, call, c/ capitalize/ text/ squ/ single

ELEMENT Other.orders, statord/ text/ squ/ single

EXPLAIN Enter information about any other orders for this same title.

ELEMENT Vendor, vend/ text/ squ/ index/ single

EXPLAIN This element is used primarily for the approval plans,

EXPLAIN most likely only the Taylor-Carlisle plan which announces

EXPLAIN titles in advance of publication. Most entries of this type

EXPLAIN probably will be handled by processing a batch file.

EXPLAIN See also the next 2 elements.

ELEMENT Announced, date.announced/ date/ single

EXPLAIN Enter the announcement date from the approval slip.

ELEMENT Available, date.due/ date/ index/ single

EXPLAIN Enter the availability date (expected date of publication)

EXPLAIN from the approval slip.

ELEMENT Received/ date/ single/ index

EXPLAIN Enter the date the item is received. This element may prove

EXPLAIN useful in the future for generating a new books list.

ELEMENT Notes/ text/ squ/ single/ priv-tag 1

EXPLAIN Enter any additional information or notes to explain the
EXPLAIN status of the order.
END

APPENDIX B. SCIENCE ORDERS FILE DEFINITION

This is the SCIENCE ORDERS file definition as it existed in the SPIRES subfile FILEDEF on 12/14/86. Most of the code was generated by the file definer listed in the previous appendix. Some changes had to be made by directly editing the file definition, as the priv tags were not implemented by the file definer at the time of generation. Subsequent editing was done to change user access privileges. Hence there is some slight departure of this file definition from the original file definer.

```
FILE = SCIBASE: SCIENCE. ORDERS;
COMMENTS = Generated by FILE DEFINER 05/27/85;
COMMENTS = NOS priv tags added manually, 5/29/85;
COMMENTS = userid links and entry commands added manually 9/11/85;
COMMENTS = linked FOLTIN 4/3/86;
COMMENTS = LINKED ACQPJM 10/3/86;
COMMENTS = linked GODAIR (Warmann) 12/12/86;
COMMENTS = added $SQU INPROC to ACCOUNT element 12/14/86;
AUTHOR = H. M. KRIZ, Sci-Tech Department, Newman Library, 6354;
DEFDATE = WED. MAY 29, 1985;
MODDATE = FRI. DEC. 12, 1986;
MODTIME = 08:45:04;
RECORD-NAME = REC01;
  REMOVED;
  SLOT;
    SLOT-NAME = ID;
  FIXED;
    ELEM = DATE. ADDED;
      OCCURS = 1;
      LENGTH = 4;
      INPROC = $DATE/ $GEN. DATE(ADD);
      OUTPROC = $DATE. OUT;
      ALIASES = DA;
      ELEMINFO;
        DESCRIPTION = " This system supplied date can be taken as the
date the order";
        DESCRIPTION = was forwarded to Acquisitions. For approval
plans, see the;
        DESCRIPTION = elements ANNOUNCED and AVAILABLE for more
information.;
      ELEM = DATE. UPDATED;
        OCCURS = 1;
        LENGTH = 4;
        INPROC = $DATE/ $GEN. DATE(UPD);
        OUTPROC = $DATE. OUT;
        ALIASES = DU;
        ELEMINFO;
```

```

        DESCRIPTION = " This system supplied date tells the last time
a record was";
        DESCRIPTION = revised.;
REQUIRED;
    ELEM = TITLE;
        OCCURS = 1;
        INPROC = $SQU;
        ALIASES = TI, T;
        ELEMINFO;
        DESCRIPTION = " Enter the ISBN number at the end of the
title.";
        DESCRIPTION = Enter the title as determined from the OCLC
database, or in;
        DESCRIPTION = the case of approval plans, as supplied by the
vendor. Other-;
        DESCRIPTION = wise enter the title as supplied by the
publisher.;
OPTIONAL;
    ELEM = AUTHOR;
        OCCURS = 1;
        INPROC = $SQU;
        ALIASES = AU, A;
        ELEMINFO;
        DESCRIPTION = " This includes personal authors, corporate
bodies, and";
        DESCRIPTION = names of conferences.;
    ELEM = PLACE;
        OCCURS = 1;
        INPROC = $SQU;
    ELEM = PUBLISHER;
        OCCURS = 1;
        INPROC = $SQU;
        ALIASES = PU, PUB;
    ELEM = EDITION;
        OCCURS = 1;
        INPROC = $SQU;
        ALIASES = ED;
    ELEM = PUBDATE;
        OCCURS = 1;
        INPROC = $SQU;
        ALIASES = PD, YEAR;
        ELEMINFO;
        DESCRIPTION = " Any text may be entered for date ranges,
etc.";
    ELEM = VOLUMES;
        OCCURS = 1;
        INPROC = $MAX.LEN(3);
        ALIASES = VOLS;
    ELEM = LISTPRICE;
        OCCURS = 1;
        ALIASES = PRICE;
    ELEM = COPIES;
        OCCURS = 1;
        LENGTH = 4;

```

```

        INPROC = $INT(4);
        OUTPROC = $INT.OUT;
ELEM = RUSHORD;
        OCCURS = 1;
        LENGTH = 4;
        INPROC = $INT(4);
        OUTPROC = $INT.OUT;
        ELEMINFO;
        DESCRIPTION = " Enter the numeral 1 for a RUSH ORDER.
Otherwise enter";
        DESCRIPTION = nothing.;
ELEM = RUSHCATA;
        OCCURS = 1;
        LENGTH = 4;
        INPROC = $INT(4);
        OUTPROC = $INT.OUT;
        ELEMINFO;
        DESCRIPTION = " Enter the numeral 1 if the item is to be RUSH
CATALOGED.";
        DESCRIPTION = Otherwise enter nothing.;
ELEM = INSTRUCTIONS;
        OCCURS = 1;
        INPROC = $SQU;
        ALIASES = SPECIALINST;
        PRIV-TAG = 1;
        ELEMINFO;
        DESCRIPTION = " Enter here any special instructions for
processing the";
        DESCRIPTION = item after receipt, such as CATALOG REF/SCI, or
SEE REQUESTOR;
        DESCRIPTION = before cataloging.;
ELEM = REQUESTOR;
        OCCURS = 1;
        INPROC = $SQU;
        ALIASES = REQ;
        PRIV-TAG = 1;
        ELEMINFO;
        DESCRIPTION = " Enter as LIBRARIAN/OTHER REQUESTOR. The
Librarian's name";
        DESCRIPTION = must come first to allow consistency in
subsequent processing.;
        DESCRIPTION = The '/' punctuation is required as the separator
between;
        DESCRIPTION = the librarian's name and any other names.;
ELEM = ACCOUNT;
        OCCURS = 1;
        INPROC = $CAP/ $SQU/ $MAX.LEN(6);
        ALIASES = AC, ACNT, FUND, FD;
        ELEMINFO;
        DESCRIPTION = " Enter the standard departmental account
acronyms.";
ELEM = CALLNO;
        OCCURS = 1;
        INPROC = $CAP/ $SQU;

```

```

    ALIASES = CALL, C;
    ELEM = OTHER.ORDERS;
    OCCURS = 1;
    INPROC = $SQU;
    ALIASES = STATORD;
    ELEMINFO;
    DESCRIPTION = " Enter information about any other orders for
this same title.";
    ELEM = VENDOR;
    OCCURS = 1;
    INPROC = $SQU;
    ALIASES = VEND;
    ELEMINFO;
    DESCRIPTION = " This element is used primarily for the
approval plans,";
    DESCRIPTION = most likely only the Taylor-Carlisle plan which
announces;
    DESCRIPTION = titles in advance of publication. Most entries
of this type;
    DESCRIPTION = probably will be handled by processing a batch
file.;
    DESCRIPTION = See also the next 2 elements.;
    ELEM = ANNOUNCED;
    OCCURS = 1;
    LENGTH = 4;
    INPROC = $DATE;
    OUTPROC = $DATE.OUT;
    ALIASES = DATE.ANNOUNCED;
    ELEMINFO;
    DESCRIPTION = " Enter the announcement date from the approval
slip.";
    ELEM = AVAILABLE;
    OCCURS = 1;
    LENGTH = 4;
    INPROC = $DATE;
    OUTPROC = $DATE.OUT;
    ALIASES = DATE.DUE;
    ELEMINFO;
    DESCRIPTION = " Enter the availability date (expected date of
publication)";
    DESCRIPTION = from the approval slip.;
    ELEM = RECEIVED;
    OCCURS = 1;
    LENGTH = 4;
    INPROC = $DATE;
    OUTPROC = $DATE.OUT;
    ELEMINFO;
    DESCRIPTION = " Enter the date the item is received. This
element may prove";
    DESCRIPTION = useful in the future for generating a new books
list.;
    ELEM = NOTES;
    OCCURS = 1;
    INPROC = $SQU;

```

```

        PRIV-TAG = 1;
        ELEMINFO;
        DESCRIPTION = "  Enter any additional information or notes to
explain the";
        DESCRIPTION = status of the order.;
RECORD-NAME = ZIN02;
REQUIRED;
    KEY = DATE.ADDED;
    INPROC = $DATE;
    OUTPROC = $DATE.OUT;
OPTIONAL;
    ELEM = POINTER;
    TYPE = LCTR;
    INPROC = $HEX;
    OUTPROC = $HEX.OUT;
RECORD-NAME = ZIN03;
COMBINE = ZIN02;
REQUIRED;
    KEY = TITLE;
OPTIONAL;
    ELEM = POINTER;
    TYPE = LCTR;
    INPROC = $HEX;
    OUTPROC = $HEX.OUT;
RECORD-NAME = ZIN04;
COMBINE = ZIN02;
REQUIRED;
    KEY = AUTHOR;
OPTIONAL;
    ELEM = POINTER;
    TYPE = LCTR;
    INPROC = $HEX;
    OUTPROC = $HEX.OUT;
RECORD-NAME = ZIN05;
COMBINE = ZIN02;
REQUIRED;
    KEY = REQUESTOR;
OPTIONAL;
    ELEM = POINTER;
    TYPE = LCTR;
    INPROC = $HEX;
    OUTPROC = $HEX.OUT;
RECORD-NAME = ZIN06;
COMBINE = ZIN02;
REQUIRED;
    KEY = ACCOUNT;
OPTIONAL;
    ELEM = POINTER;
    TYPE = LCTR;
    INPROC = $HEX;
    OUTPROC = $HEX.OUT;
RECORD-NAME = ZIN07;
COMBINE = ZIN02;
REQUIRED;

```

```

    KEY = VENDOR;
OPTIONAL;
    ELEM = POINTER;
    TYPE = LCTR;
    INPROC = $HEX;
    OUTPROC = $HEX.OUT;
RECORD-NAME = ZIN08;
    COMBINE = ZIN02;
REQUIRED;
    KEY = AVAILABLE;
    INPROC = $DATE;
    OUTPROC = $DATE.OUT;
OPTIONAL;
    ELEM = POINTER;
    TYPE = LCTR;
    INPROC = $HEX;
    OUTPROC = $HEX.OUT;
RECORD-NAME = ZIN09;
    COMBINE = ZIN02;
REQUIRED;
    KEY = RECEIVED;
    INPROC = $DATE;
    OUTPROC = $DATE.OUT;
OPTIONAL;
    ELEM = POINTER;
    TYPE = LCTR;
    INPROC = $HEX;
    OUTPROC = $HEX.OUT;
GOALREC-NAME = REC01;
    PTR-ELEM = POINTER;
    EXTERNAL-NAME = TITLE;
    PASSPROC = $PASS.LCTR;
INDEX-NAME = ZIN02;
    SEARCHTERMS = DATE.ADDED, DA;
    SEARCHPROC = $DATE(TRUNC);
    PASSPROC = $PASS.ELEM('DATE.ADDED',NUMERIC)/ $NULL;
    PTR-GROUP = POINTER;
INDEX-NAME = ZIN03;
    SEARCHTERMS = TITLE, TI, T;
    SEARCHPROC = $WORD(SEARCH)/ $EXCLUDE(COMMON.WORDS);
    PASSPROC = $PASS.ELEM('TITLE',2)/ $WORD(PASS)/
$EXCLUDE(COMMON.WORDS)/ $NULL;
    PTR-GROUP = POINTER;
INDEX-NAME = ZIN04;
    SEARCHTERMS = AUTHOR, AU, A;
    SEARCHPROC = $WORD(SEARCH)/ $EXCLUDE(COMMON.WORDS);
    PASSPROC = $PASS.ELEM('AUTHOR',2)/ $WORD(PASS)/
$EXCLUDE(COMMON.WORDS)/ $NULL;
    PTR-GROUP = POINTER;
INDEX-NAME = ZIN05;
    SEARCHTERMS = REQUESTOR, REQ;
    PRIV-TAG = 1;
    SEARCHPROC = $WORD;
    PASSPROC = $PASS.ELEM('REQUESTOR',2)/ $WORD(PASS)/ $NULL;

```

```

PTR-GROUP = POINTER;
INDEX-NAME = ZIN06;
SEARCHTERMS = ACCOUNT, AC, ACNT, FUND, FD;
SEARCHPROC = $MAX.LEN(6);
PASSPROC = $PASS.ELEM('ACCOUNT',1)/ $NULL;
PTR-GROUP = POINTER;
INDEX-NAME = ZIN07;
SEARCHTERMS = VENDOR, VEND;
SEARCHPROC = $CAP;
PASSPROC = $PASS.ELEM('VENDOR',1)/ $NULL;
PTR-GROUP = POINTER;
INDEX-NAME = ZIN08;
SEARCHTERMS = AVAILABLE, DATE.DUE;
SEARCHPROC = $DATE(TRUNC);
PASSPROC = $PASS.ELEM('AVAILABLE',NUMERIC)/ $NULL;
PTR-GROUP = POINTER;
INDEX-NAME = ZIN09;
SEARCHTERMS = RECEIVED;
SEARCHPROC = $DATE(TRUNC);
PASSPROC = $PASS.ELEM('RECEIVED',NUMERIC)/ $NULL;
PTR-GROUP = POINTER;
SUBFILE-NAME = SCIENCE ORDERS;
EXP = " This database lists titles on order or scheduled for";
EXP = delivery through approval plans to the Science & Technology;
EXP = Department, Newman Library. The file is purged periodically;;
EXP = usually after printing of the monthly list of Selected New;
EXP = books. No attempt is made to track the progress of an order;
EXP = through the Library's Acquisitions and Cataloging
Departments.;
EXP = Please check VTLS or the Library In-Process List for the most;
EXP = recent information about an order, or ask a Reference;
EXP = librarian for assistance.;
GOAL-RECORD = REC01;
ACCOUNTS = SCIBASE;
SELECT-COMMAND = set format $prompt;
SELECT-COMMAND = set xeq stprotos;
SELECT-COMMAND = ..ordersentrypriv;
GOAL-RECORD = REC01;
ACCOUNTS = SCIP, SCIS, SCISPI, SCIV, RENO, KOK, KRIZ, SCICO,
REFER, TRAN, SCITECH, ACQJPM, RECSEC, RECSUP, GHERMAN, FOLTIN, GODAIR;
SECURE-SWITCHES = 3;
SELECT-COMMAND = Set format $prompt;
SELECT-COMMAND = set xeq stprotos;
SELECT-COMMAND = ..ordersentrypriv;
GOAL-RECORD = REC01;
ACCOUNTS = PUBLIC;
SECURE-SWITCHES = 3;
CONSTRAINT = 1;
NOUPDATE = 1;
NOSEARCH = 1;
SELECT-COMMAND = Set format $prompt;
SELECT-COMMAND = set xeq stprotos;
SELECT-COMMAND = ..ordersentrypub;

```

APPENDIX C. CREATING THE SCIENCE ORDERS DATABASE

C.1 PROCEDURES OUTLINE

1. Create the file definer in a CMS file.
2. Generate the file definition from the definer using the FILE DEFINER subsystem.
3. Add the file definition to the FILEDEF subfile.
4. Compile the file definition.

C.2 DETAILED PROCEDURE

1. XEDIT a CMS file to enter the description of the database elements using the FILE DEFINER language. Save this file as ORDERS DEFINER A. For appearance and readability, this file should contain both upper and lower case letters. Use care in editing any changes since XEDIT will default to all uppercase for a filetype other than SCRIPT or MEMO.
2. SPIRES
3. ENTER FILE DEFINER
4. SET ACTIVE ORDERS DEFINER A
Spires will then use ORDERS DEFINER A for the input file.
5. INPUT ACTIVE
On this command, SPIRES will read the contents of the active file (ORDERS DEFINER A) into the file definer's input file. The input lines are echoed to the screen. If the input fails, as indicated by an abort message, reedit ORDERS DEFINER A to correct any errors. Repeat the above procedures until the active file is input successfully.
6. SET ACTIVE ORDERS FILEDEF A
This changes the active file to a new, as yet nonexistent, file named ORDERS FILEDEF A, which will be used to accept the output from the file definer. If the SPIRES active file were not changed at this point, writing to the original active file would write over the original file definer. Any changes to element descriptions would then require complete reconstruction of the original definer file.
7. GENERATE
This command causes the file definer program to generate the file definition and place it in the active file (ORDERS FILEDEF A). If there is already a file definition from a previous attempt, the system will ask O. K. TO CLEAR? to which the reply should be Y. SPIRES will

not write to an existing file unless it is instructed to do so, thereby preventing accidental loss of data.

8. The return of the file definer prompt indicates a file definition was generated.
9. **LIST**
This command will show the file definition on the screen. The FL command will show that ORDERS DEFINER A and ORDERS FILEDEF A are CMS files which can be used in the future.
10. **SPIRES**
It is necessary to return to SPIRES from the FILE DEFINER system before proceeding.
11. **SELECT FILEDEF**
This command selects the SPIRES system subfile FILEDEF which is a database of file definitions.
12. **ADDUPD**
This command updates the SPIRES system subfile FILEDEF by adding the SCIENCE ORDERS file definition. Specifically, the contents of the active file are added to FILEDEF. UPD is not used on the first entry of a file definition, but it must be used for all changes.
13. **SPICOMP**
This command calls the compiler.
14. **COMPILE SCIENCE.ORDERS**
This compiles the file definition and establishes several CMS files used by the database. It is possible for a file definition which generated successfully to fail to compile. When using the file definition language, a 2-part subfile name requires a "." between the 2 parts. The "." must be used in the compile process. The select command does not require the "." once the database is compiled.
15. **SPIRES**
This invokes SPIRES following compilation.
16. **SELECT SCIENCE ORDERS**
This selects the new database.

APPENDIX D. SHARED PUBLIC ACCESS

Placing the SPIRES data files on a 200 disk provides security against the accidental erasure possible when the database files are on the default A disk. It also allows sharing of the database among several users and the general public through modification of the file definition if this is desired. This procedure was followed for SCIENCE ORDERS. This appendix describes the procedure for establishing the SPIRES database on a 200 disk. The procedures assume the database has already been established on the A disk.

D.1 ESTABLISHING THE 200 DISK

When a 200 disk (direct access storage device) is first assigned it must be accessed, and a password must be assigned. Log on to the userid which owns the disk.

QDISK DASD to get number of the disk to
be shared, e.g. 200

ACC 200 C to access the disk and call it
the C disk for this session. If
there is already a C disk,
use another letter.

copy all the SPIRES database files to the C disk

As a precaution, to avoid confusion, rename the SPIRES files on the A disk after they have been copied to the 200 disk. They should be saved as a precaution so the data is not lost by accident during these changes.

SPIRES must now be told the new location of the data files by using the FILELOC subfile. A location record is written in the active file and added to the FILELOC subfile giving the virtual address of the files, e.g. 200.

X ORDERS FILELOC

This file should contain the following 3 lines

FILE = [userid]:SCIENCE.ORDERS; in other words, the full name of the
SPIRES file as used in the file definer.

VADDR = 200;

FILETYPE = *; This indicates all files for the database will
be on the 200 disk.

This file definition is added to the FILELOC subfile, using the following series of commands:

SPIRES
SELECT FILELOC
SET ACTIVE ORDERS FILELOC
ADD

It is necessary to change the mode of access to the 200 disk, which originally is set up as read/write. It cannot be left this way because SPIRES would not be able to get write access if the owner userid was logged on. Therefore, the userid's access is changed to read only using the following procedure:

DIRM MD

Fill in the appropriate parts of the menu with

ACCESS: RR

READ PASSWORD: ALL This means read security is provided only
 through SPIRES.

A write password and read/write password must also be assigned, they can be the same.

In addition, SPIRES must be given the write password for the 200 disk. While in SPIRES issue the command

SET PASSWORD 200 [password]

Every database on this 200 disk will use the same password.

Recompile the file definition using the commands

SPICOMP
RECOMPILE [USERID]:SCIENCE.ORDERS

Detach the 200 disk so it is not linked when the owner userid logs on by issuing the command:

DETACH 200

This completes the transfer of the database to the 200 disk.

The 200 disk will not show on a q disk unless the database on the 200 disk has been selected. However, the owner userid can always link, even without the password by issuing

LINK * 200 200 WR

although this will fail if SPIRES already has a link.

D.2 SHARING THE DATABASE

To make a shared database involves only updating the subfile section of the file definition. It does not require a recompile unless there are changes in the element definitions. Security switches and priv tags do require a recompile, but the data does not need to be unloaded and reloaded.

D.2.1 Notes

- Note that the GOAL element associates with the most recent subfile name.
- If an indexed element is to be kept secure, the NOSEARCH option must be specified.
- Always keep a copy of the working filedef.
- Zapping the file will remove it from the 200 disk.

Protocols must be in a public or shared protocol subfile to be used by the shared database. In the protocol subfile, it is useful to use CLEAR SELECT after SET XEQ to prevent anyone from selecting the protocol subfile and viewing the protocol source code.

APPENDIX E. ORFLOAD PROTOCOL

This is the protocol which translates the ORF image files (ROUTE1 DATA) into a form readable by SPIRES for batch loading into SCIENCE ORDERS. It is run from the active file.

```
! SET NO ECHO
clear
*
*
*               ORFS REFORMAT PROTOCOL
*
*   This is a protocol to convert the most recently assembled version of
*   ORFSmdd DATA to the file ORFLOAD DATA for processing into the
*   SCIENCE ORDERS database.
*
*   Any existing ORFLOAD DATA file was erased when you issued the
*   ORFLOAD command.
*
*                                     (Press <RETURN> to continue)
pause
clear
*
* BEFORE CONTINUING, YOU SHOULD HAVE DONE THE FOLLOWING:
* 1.) Received on this userid all order files sent to Acquisitions for
*     the current period.
* 2.) PUT the individual order files into a single file named
*     ORFSmdd DATA, where mm is the month, and dd is the day on which
*     the orders were processed.
* 3.) To reduce the chance of error,
*     You will be prompted for the exact fileid of the file to be
*     processed. You may wish to write it down now.
* 4.) Erased any previously existing ORFSmdd DATA file to avoid
*     confusion and save disk space. Normally this is done after
*     the successful loading of the data into SPIRES during the
*     previous week.
*
*++ QUERY1
ask upper prompt = 'Are all preparations complete? (Y/N)'
if $ask = Y: jump GO1
if $ask = N: jump NOGO
* Y or N please
jump query1
*++ NOGO
clear
*
*
*
*****
```

```

*       Please complete preparations before running this protocol
*
*       Returning to SPIRES command level
*****
jump END.END
++ GO1
*
*       Enter <fn> of file to be processed in the form ORFSmdd
*       (Enter Q to quit)
ask upper
let orfinpt = $ask
if #orfinpt = Q then jump NOGO
*
/* The file #orfinpt DATA will be processed
*
++ QUERY2
ask upper prompt = '       Is the fileid correct? (y/n)'
if $ask = Y: jump GO2
if $ask = N: jump GO1
* Y or N please
jump gol
++ GO2
/set active #orfinpt data (idata
set active ORFLOAD DATA (odata
* Acquisitions file is being converted to ORFLOAD DATA for input to
* the SCIENCE ORDERS database. Please be patient.
- Try measuring length of file and estimating time-----
set wdsr 1
let count = 0
++ BEGIN
let count = #count + 1
- Need to specify blank lines or program will skip and lose count.
WDSR END = "jump END"
let line1 = $ask
wdsr null = "set ask = ' '"
let line2 = $ask
wdsr
let line3 = $ask
wdsr null = "set ask = ' '"
let line4 = $ask
wdsr
let line5 = $ask
wdsr
let line6 = $ask
wdsr
let line7 = $ask
wdsr null = "set ask = ' '"
let line8 = $ask
wdsr
let line9 = $ask
wdsr
let line10 = $ask
wdsr
let line11 = $ask

```

```

wdsr
let line12 = $ask
wdsr
let line13 = $ask
- line14 is blank and indicates end of record
wdsr null = "jump batch.write"
-----
-----
++ BATCH.WRITE
-          a routine to concatenate and clean strings before writing
-----
- CLEAN THE STRINGS
let title = $rightstr(#line1,3)#line2$substr(#line3,1,54)
let title = $break(#title,'|')
let author = $rightstr(#line3,59)#line4
let author = $break(#author,'|')
let place = $substr(#line5,25,29)
let place = $break(#place,'|')
let publisher = $rightstr(#line5,58)$leftstr(#line6,37)
let publisher = $break(#publisher,'|')
let edition = $rightstr(#line6,41)$substr(#line7,1,3)
let edition = $break(#edition,'|')
let pubdate = $substr(#line7,8,10)
let pubdate = $break(#pubdate,'|')
let vols = $substr(#line7,22,2)
let listprice = $substr(#line7,27,7)
let copies = $substr(#line7,37,2)
let rushord = $substr(#line9,7,1)
let rushcata = $substr(#line9,11,1)
let specialinst = $substr(#line9,15,60)
let specialinst = $break(#specialinst,'|')
let requestor = $rightstr(#line9,78)$leftstr(#line10,33)
let account = $substr(#line10,36,4)
let callno = $substr(#line11,7,40)
let statord = $rstr(#line11,50)$lstr(#line12,10)
- WRITE THE FILE, if first character is blank, no entry, don't write
wdse add;
-----
if $size(#title) > 124 then jump split.title
-----
/wdse t = #title;
++ WRITE.AUTHOR
if $lstr(#author,1) ~= ' ': /wdse author = #author;
if $lstr(#place,1) ~= ' ': /wdse place = #place;
/wdse publisher = #publisher;
if $lstr(#edition,1) ~= ' ': /wdse edition = #edition;
/wdse pubdate = #pubdate;
if $lstr(#vols,1) ~= ' ': /wdse vols = #vols;
/wdse listprice = #listprice;
if $lstr(#copies,1) ~= '1': /wdse copies = #copies;
if $lstr(#rushord,1) ~= ' ': /wdse rushord = #rushord;
if $lstr(#rushcata,1) ~= ' ': /wdse rushcata = #rushcata;
if $lstr(#specialinst,1) ~= ' ': /wdse specialinst = #specialinst;

```

```

let requestor = $change(#requestor, '/', ', ' )
/wdse requestor = #requestor;
/wdse account = #account;
if $lstr(#callno,1) ~= ' ': /wdse callno = #callno;
if $lstr(#statord,1) ~= ' ': /wdse statord = #statord;
/wdse ;
/* Record #count processed.
- return to beginning to read the next record
jump begin
-----
++ END
*
let count = #count - 1
/* #count records have been processed.
*****
*
*           The file ORFLOAD DATA is ready for loading
*           into the SCIENCE ORDERS database.
*
*****
++ END.END
set active active file
set format $prompt
* Returning to SPIRES command level
* Active file set to <fn> <ft> ACTIVE FILE
* Format set to $prompt
*
RETURN
-----
---SUBROUTINES
-----
++ SPLIT. TITLE
let title1 = $lsub(#title, ' ')
let title2 = $rsub(#title, ' ')
if $SIZE(#TITLE1) < 55 THEN JUMP BUILD.TITLE1
else jump split.title.done
++ BUILD. TITLE1
let title1 = #title1 ' $lsub(#title2, ' ')
let title2 = $rsub(#title2, ' ')
if $size(#title1) < 55 then jump BUILD.TITLE1
else jump split.title.done
++ SPLIT. TITLE. DONE
/wdse t = #title1
/wdse #title2;
jump write.author
-----

```

APPENDIX F. ROUTE1 DATA FILE

On the next page are two sample records from the ROUTE1 DATA file created by the Acquisitions Department's data entry system. It is a file of this type which is translated by the ORFLOAD PROTOCOL into a form readable by the SPIBILD batch loader. The format for ROUTE1 DATA is dictated by the input requirements of the mainframe based acquisitions system which has been in use at Virginia Tech for at least a decade. The details of this file have no significance outside of the Virginia Tech environment.

Each record in ROUTE1 DATA consists of 13 lines of data, with 1 blank line separating each record. Some data is in fixed length fields. Other information is set off by delimiters. The details are more fully described in THE ACQUISITIONS SYSTEM: Reference Department's Procedure by Bill Dougherty & Betty McClellan, Library Systems Office, September, 1984.

On the second page following is the output from ORFLOAD PROTOCOL after processing the sample ROUTE1 DATA. The trailing blanks in each line are eliminated when the records are loaded through SPIBILD. They could just as well have been eliminated within the ORFLOAD PROTOCOL, but this would introduce unnecessary processing as the blanks will be eliminated due to the SQUEEZE attribute in the file definition for each element of the record.

OB POST-YIELD FRACTURE MECHANICS. ISBN 0853342768. |

> |A LATZKO, D. G. H. |
> |CBNEW YORK, | |CCEELSEVIER APPLIED SCIEN
CE, | |CA2ND, |
> |CD1984. | |DB |DC0108.00|DD1 |FB |F
C |GA |GB |GC | |HAGR
AYBEAL |HBENGR|DAVVPOS
>FM*|IBTA409 P67 NED |IC
|JA00 |JB00 |JE
> |JC4|JD1|JF ~

OB INTEGRATED MATERIALS HANDLING IN MANUFACTURING. ISBN 0903608936. |

> |A MULLER, WILLI. |
> |CBNEW YORK, | |CCSPRINGER-VERLAG, |
|CA |
> |CD1985. | |DB |DC0030.00|DD1 |FB |F
C |GA |GB |GC | |HAGR
AYBEAL |HBENGR|DAVVINT
>MHI|IB |IC
|JA00 |JB00 |JE
> |JC4|JD1|JF ~

```

add;
t = POST-YIELD FRACTURE MECHANICS. ISBN 0853342768.;
author = LATZKO, D.G.H.;
place = NEW YORK;;
publisher = ELSEVIER APPLIED SCIENCE;;
edition = 2ND;;
pubdate = 1984.;
listprice = 0108.00;
requestor = GRAYBEAL;
account = ENGR;
callno = TA409 P67 NED;
;
add;
t = INTEGRATED MATERIALS HANDLING IN MANUFACTURING. ISBN 0903608936.;
author = MULLER, WILLI.;
place = NEW YORK;;
publisher = SPRINGER-VERLAG;;
pubdate = 1985.;
listprice = 0030.00;
requestor = GRAYBEAL;
account = ENGR;
;

```

APPENDIX G. NEWALL PROTOCOL

This is the protocol which produces the list of all new books received during a given month. See the chapter on the New Books List for details.

```
! set no echo
clear
*
*               ALL NEW BOOKS
*
* This protocol finds all titles received during a given month.
* You will be prompted for the month and year. Six (6) copies will
* be printed. The resulting list is for distribution to librarians in
* the Science & Technology Department.
*
++ BEGIN.QUERY
ask upper prompt "Do you wish to continue? y/n"
if $ask = "Y" then jump START
if $ask = "N" then jump QUIT
* Enter y or n please
jump BEGIN.QUERY
++ START
clear
clear elements
++ G01
*
*
* Enter month and year covered by the New Books List (e.g. January, 1985)
ask
let period = $ask
/* New books list is for the month of #period
* Is this correct?(y/n)
++ G01.1
ask upper
IF $ASK = "Y": JUMP G02
IF $ASK = "N": * REENTER DATE
then jump G01
* y or n please
jump G01.1
++ G02
/* Finding all titles received in #period
/FIND RECEIVED #PERIOD
* Sequencing titles and call numbers
seq call title
* Setting format for printed output with 6 copies
SETP 3800L10 CO 6
* Printer tag set.
*
```

```

* Now setting format, patience please.
let listtitle = "Books received: " #period
SET FORMAT $REPORT id(1,5) da(+1,8,heading='Date//ordered)
SET FORMAT * + C(+1,2,heading="")
SET FORMAT * + Title(+3,47,indent=-2) Author(+1,20,indent=-2)
SET FORMAT * + Publisher(+1,20) Year(+2,4)
SET FORMAT * + Requestor(+1,10,indent=-1) ac(+1,4,heading='ACNT')
SET FORMAT * IN ROW +1 vend(+27,15,heading='') available(+2,8,heading='')
SET FORMAT * skip 1
SET FORMAT * GROUPED BY C
SET FORMAT * AT START OF PAGE, print #listtitle// //$date '      page ' $pageno
SET FORMAT * define value title.count as count title
SET FORMAT * AT END OF REPORT, print(left) '-----'
SET FORMAT * in row 4, print 'Number of titles: ' @title.count
SET FORMAT * OPTIONS lines = 72
SET FORMAT * AT END OF GROUP C, PRINT(CENTER) '-----'
SET FORMAT * Skip 1
* Format set for list: #listtitle
*      List being written to NEWBOOKS FILE
set active newbooks file
in active clear type
* New books list being printed
PRINT NEWBOOKS FILE (CC
clear
*
*
*      *****
/*      New books list printed for #period
*      *****
*
*
*
++ QUIT
set active active file
set format $prompt
SETP 3800P6 PRE CO 1
*      Returning to SPIRES command level
*      Active file set to ACTIVE FILE
*      Format set to $PROMPT
*      Printer set to Prestige 12 pitch
*
++ EXIT
* RETURNING TO SPIRES COMMAND MODE.

```

APPENDIX H. NEWSELCT PROTOCOL

This is the protocol which produces the list of SELECTED NEW BOOKS. See the chapter on the New Books List.

```
! set no echo
clear
*
*                               SELECTED NEW BOOKS
*
* This protocol produces the Selected New Books list published each
* month by the Science & Technology Department. It should be invoked
* only after using the NEWALL protocol and editing the contents of the
* database to remove those records which are not wanted on this
* selected list.
*
++ BEGIN. QUERY
ask upper prompt "Do you wish to continue? y/n"
if $ask = "Y" then jump START
if $ask = "N" then jump QUIT
* Enter y or n please
jump BEGIN. QUERY
++ START
clear
clear elements
++ G01
* Enter month and year covered by the New Books List (e.g. January, 1985)
ask
let period = $ask
/* New books list is for the month of #period
* Is this correct?(y/n)
++ G01.1
ask upper
IF $ASK = "Y": JUMP G02
IF $ASK = "N": * REENTER DATE
then jump G01
* y or n please
jump G01.1
++ G02
/* Finding all titles received in #period
/FIND RECEIVED #PERIOD
* Sequencing titles and call numbers
seq call title
++ COPIES.CHOICE
* Enter the number 1 to print 1 test copy to verify content and form
* Enter the number 2 to print 80 copies for distribution
ask
++ COPIES.VALID
```

```

IF $ASK = "1" : LET COPYNUM = "1"
THEN JUMP COPIES.CHECK
IF $ASK = "2" : LET COPYNUM = "80"
THEN JUMP COPIES.CHECK
else * 1 or 2 please
else jump COPIES.CHOICE
++ COPIES.CHECK
/* You selected #copynum copies. Is this correct? (y/n)
ask upper
if $ask = Y: jump COPIES.GO
if $ask = N: * Reenter copies
then jump COPIES.CHOICE
* y or n please
jump COPIES.CHECK
++COPIES.GO
if #copynum = "1" : jump ONE.COPY
if #copynum = "80" : jump EIGHTY.COPIES
* 1 or 2 please
jump COPIES.CHOICE
++ ONE.COPY
setp 3800p10 co 1
jump SET.FORMAT
++ EIGHTY.COPIES
setp 3800p10 co 80
jump SET.FORMAT
++SET.FORMAT
* Setting format for printed output
SET FORMAT $REPORT c(1,5,trunc,heading = "LC//Class")
SET FORMAT * + TITLE(+3,60,indent=-3) Author(+2,20,indent=-2)
SET FORMAT * + Publisher(+2,20,trunc) Year(+2,4,trunc)
SET FORMAT * skip 1
SET FORMAT * GROUPED BY c
SET FORMAT * AT START OF PAGE, PRINT //'SELECTED NEW BOOKS'//'page ' $pageno
SET FORMAT * IN ROW 2, PRINT(CENTER) 'for ' #period
SET FORMAT * IN ROW 3, PRINT //'Sci-Tech Department'//$date
SET FORMAT * IN ROW 4, PRINT(CENTER) 'Newman Library'
SET FORMAT * DEFINE value title.count as count title
SET FORMAT * AT END of report, print(left) '-----'
SET FORMAT * in row 4, print 'Number of titles: ' @title.count
SET FORMAT * OPTIONS lines = 100
SET FORMAT * OPTIONS indent = 5
SET FORMAT * AT END OF GROUP call, print(center) '-----'
SET FORMAT * skip 1
clear
*
*
* Format set, report being printed.
* List being written to NEWBOOKS FILE
set active newbooks file
IN ACTIVE CLEAR TYPE
* New books list being printed
PRINT NEWBOOKS FILE (CC
clear
*

```

```

*
* *****
/*      New books list printed for #period
*      *****
*
*
*
++ QUIT
set active active file
set format $prompt
SETP 3800P6 PRE CO 1
*      RETURNING TO SPIRES COMMAND LEVEL
*      Active file set to ACTIVE FILE
*      Format set to $PROMPT
*      Printer set to Prestige 12 pitch
*
RETURN

```

APPENDIX I. ..PRINTOUT PROTOCOL

This protocol produces several different standard reports for private users of the SCIENCE ORDERS database. It is entered as a record in the public subfile STPROTOS. It may be invoked from any userid by issuing the command `..printout` It will not run to completion for public mode users as it uses the REQUESTOR element which is hidden in public mode.

```
! set no echo
let formatflag = 'FF0'
-
- ***** PRINT A MENU AND OPENING SCREEN
++ PRINTOUT
clear
*
*                               SCIENCE ORDERS PRINTOUTS
*
* Generate a printed report from the SCIENCE ORDERS database
* by selecting from the menu. These procedures will NOT run to
* completion on public userids.
*
*       1.  General List Formatter
*           (Sets a format and prints a list for the most recently
*            obtained search result)
*
* The following routines conduct standard searches and print a report:
*
*       2.  All orders or RUSH orders for a requestor
*       3.  Subscriptions or standing orders for a requestor
*       4.  CANCEL printout request
*
xeq proc WARNING
ask  prompt 'Enter a number from 1 to 4'
if $ask = 1 then jump GENERAL.FORMAT
if $ask = 2 then jump REQUESTOR
if $ask = 3 then jump SERIALS
if $ask = 4 then jump PRINTOUT.DONE
jump PRINTOUT
-
++ GENERAL.FORMAT
clear
*
*                               GENERAL LIST FORMATTER
*
* This procedure sets a format and produces a printout for
* an existing search result.
*
```

```

*   You will be asked whether you want the result arranged in
*   alphabetical order by title or
*   NOT arranged by this protocol and left in any order you
*   have already specified.
*
***** IF YOU HAVE NOT YET CONDUCTED YOUR SEARCH, CANCEL YOUR REQUEST
*
let listname = 'GENERAL.FORMAT'
let listsubtitle = ''
xeq proc PRINT.QUERY
-                               (On return, ask for sequence
clear
*
xeq proc SEQ.QUERY
-                               (On return, sequencing is complete
clear
*
xeq proc TITLE.QUERY
-                               (On return from TITLE.QUERY, set the format
clear
xeq proc FORMAT1
-                               (On return from setting format, do the print
xeq proc PRINT1
-                               (On return from printing, print message, ask if done
clear
*
*
*   TITLES ON ORDER
/* #listsubtitle
*
/* printed for $stack titles.
jump DONE.QUERY
-----
++ SERIALS
clear
*
*           SUBSCRIPTION AND STANDING ORDER PRINTOUT
*
*   This protocol will produce a printout of all subscriptions and
*   standing orders ordered by a specific requestor.
*
*   The list will be in alphabetical order by title.
*
*
let listname = 'SERIALS'
xeq proc CONTINUE.QUERY
-
clear
xeq proc SERIALS.SEARCH
-
clear
*
/* Your printout will contain $result titles of
/* subscriptions and standing orders for #requestor

```

```

*
xeq proc PRINT.QUERY
-
* Arranging list in alphabetical order by title.
seq title
-
/let listsubtitle = 'subscriptions and standing orders for #requestor'
-
clear
xeq proc FORMAT1
-
xeq proc PRINT1
-
clear
*
*
/* TITLES ON ORDER
/* #listsubtitle
*
/* printed for $stack titles
jump DONE.QUERY
-----
++ REQUESTOR
clear
*
*
* REQUESTOR PRINTOUT
*
* This protocol will produce a printout of titles on order
* for a specific requestor. You will be given the option of
* limiting the list to RUSH orders only.
*
* The list will be in alphabetical order by title.
*
*
*
let listname = 'REQUESTOR'
xeq proc CONTINUE.QUERY
-
clear
xeq proc REQUESTOR.QUERY
/find req #requestor
if $zresult: /* There are no orders for #requestor
then jump DONE.QUERY
clear
*
/* Your printout will contain $result titles on
/* order for the requestor #requestor.
*
-
/let listsubtitle = 'requested by #requestor'
-
xeq proc RUSH.QUERY
-
xeq proc PRINT.QUERY

```

```

-
* Arranging list in alphabetical order by title.
seq title
-
clear
xeq proc FORMAT1
-
xeq proc PRINT1
-
clear
*
*
/* TITLES ON ORDER
/* #listsubtitle
*
/* printed for $stack titles
jump DONE.QUERY
- *****
- SEARCH ROUTINES
- *****
-
++ RUSH. SEARCH
* Limiting to RUSH orders only.
for result where rushord occurs > 0
stack all END = 'jump NO.RUSH'
endfor
*
/* There are $stack RUSH orders for #requestor
*
RETURN
++ NO. RUSH
/* There are no RUSH orders for #requestor
jump DONE.QUERY
-
++ SERIALS. SEARCH
*
xeq proc REQUESTOR.QUERY
++ SERIALS.GO
find title sub or so
/and req #requestor
if $zresult then /* There are no subscriptions or standing orders for
/* #requestor
then jump DONE.QUERY
RETURN
-
- *****
- QUERIES
- *****
-
++ CONTINUE.QUERY
ask upper prompt = 'Press C to continue, M to return to printouts menu'
if $ ask = C then RETURN
if $ ask = M then jump PRINTOUT
* Enter C or M please

```

```

jump CONTINUE.QUERY
-
++ RUSH.QUERY
ask upper prompt = 'Limit to RUSH orders only? (y/n)'
/if $ask = Y then xeq proc RUSH.SEARCH
/then let listsubtitle = 'RUSH orders for #requestor'
then RETURN
if $ask = N: RETURN
* Y or N please.
jump RUSH.QUERY
-
++ PRINT.QUERY
* You may cancel your request now by typing CANCEL.
* Otherwise, obtain a printout by typing PRINT.
ask upper
if $ask = PRINT then RETURN
if $ask = CANCEL: clear
then *
then *
then *
then * *****
then * ** **
then * ** PROCEDURE CANCELED BY USER **
then * ** **
then * *****
then *
then jump DONE.QUERY
* PRINT or CANCEL please.
jump PRINT.QUERY
-
++ SEQ.QUERY
*
* Do you want this list arranged in
* alphabetical order by title? (y or n)
ask upper
if $ask = Y: * Arranging list in order by title
then seq title
then RETURN
if $ask = N: RETURN
* Y or N please
jump SEQ.QUERY
-
++ TITLE.QUERY
* The printout has the title TITLES ON ORDER
* You may enter an explanatory subtitle now, or press <Enter>
* for no subtitle.
ask prompt 'Enter a subtitle: ' NULL RETURN
/let listsubtitle = "$ask"
++ TITLE.VERIFY
*
* Your printout will be titled
* TITLES ON ORDER
/* #listsubtitle
*
* Is this correct? (y/n)

```

```

ask upper
if $ask = Y : RETURN
if $ask = N : clear
then * Try again
then jump TITLE.QUERY
* Y or N please
jump TITLE.VERIFY
-
++ REQUESTOR.QUERY
ask prompt 'Enter last name of Requestor'
/let requestor = '$ask'
++ REQ.VERIFY
/* Requestor's name is #requestor. (Respond with y or n)
ask upper
if $ask = Y: RETURN
if $ask = N: * Try again
then jump REQUESTOR.QUERY
* Y or N please
jump REQ.VERIFY
-
++ DONE.QUERY
*
*
* Press M to return to printouts menu.
*
* Press Q to quit the printouts program and return to command level.
ask upper
if $ask = M: jump PRINTOUT
if $ask = Q: jump PRINTOUT.DONE
* Enter M or Q please.
jump DONE.QUERY
-
- *****
- FORMATS
- *****
- would be useful to have a received date in the format
++ FORMAT1
if formatflag = 'FF1': jump FORMAT1.1
* Format being set. Patience please.
setp 3800L10
-
SET FORMAT $REPORT id(1,5,trunc) da(+1,8,trunc,heading='Date//ordered//received')
SET FORMAT * + Title(+3,50,indent=-2) Author(+1,20,indent=-2)
SET FORMAT * + Publisher(+1,20,trunc) Year(+2,4,trunc)
SET FORMAT * + Requestor(+1,10,indent=-1) ac(+1,4,trunc,heading='ACNT')
SET FORMAT * IN ROW +1 received (+6,8,heading='') vend(+27,15,trunc,heading='')
SET FORMAT * + available(+2,8,trunc,heading='')
SET FORMAT * skip 1
SET FORMAT * define value title.count as count title
SET FORMAT * AT END OF REPORT, print(left) '-----'
SET FORMAT * in row 4, print 'Number of titles: ' @title.count
SET FORMAT * OPTIONS lines = 72
-----
++ FORMAT1.1

```

```

SET FORMAT * AT START OF PAGE, PRINT 'TITLES ON ORDER'// // $DATE '      PAGE ' $PAGENO
/SET FORMAT * IN ROW 2, PRINT "#listsubtitle"
let formatflag = 'FF1'
*   Format set
RETURN
-
- *****
-                               PRINT ROUTINES
- *****
-
++ PRINT1
*   Writing report to OUTPUT FILE.  Please be patient.
set active OUTPUT FILE
in active clear type
*   OUTPUT FILE created.  Now sending list to the printer.
PRINT OUTPUT FILE (CC
RETURN
-
- *****
-                               MESSAGE ROUTINES
- *****
-
++ WARNING
*   >>>   This procedure will create a file named OUTPUT FILE on
*   >>>   your userid. It will overwrite any existing file with
*   >>>   the name OUTPUT FILE.
RETURN
-
- *****
-                               EXIT ROUTINE
- *****
-
++ PRINTOUT.DONE
clear
*
*
*   *****
*   Returning to SCIENCE ORDERS command level
*   *****
clear elements
set format $prompt
*                               Format reset to $prompt
set active active file
*                               Active file set to ACTIVE FILE
setp 3800p6 pre
*   Printer reset to Portrait, prestige font, 6 lpi
CLEAR XEQS
RETURN

```

APPENDIX J. COST PROTOCOL

This protocol produces a price report arranged by requestor's name and subject account for firm orders entered into SCIENCE ORDERS on a given date.

```
++ INITIALIZE
CLEAR
set active output file
clear active
++ START.QUERY
* This is a protocol which produces a price report arranged by
* requestor's name for firm orders entered into SCIENCE ORDERS
* on a given date.
*
* You will be asked to enter a particular order date.
* Once the date is entered, the report is generated automatically.
*
++ GO.QUERY
ask upper prompt = "Do you wish to continue? (y/n)"
if $ask = Y: jump BEGIN
if $ask = N: jump EXIT
* Y or N please.
jump GO.QUERY
-
clear
++ BEGIN
ask prompt 'Enter order date'
/let orderdate = '$ask'
++ DATE.VERIFY
/* Order date is #orderdate. (y or n)
ask upper
IF $ASK = Y: JUMP GO1
if $ask = N: * Try again
then jump BEGIN
* y or n please
jump DATE.VERIFY
++ GO1
/find da #orderdate
if $zresult: /* No orders were added on #orderdate
then jump EXIT
and not vendor Taylor-Carlisle
if $zresult: /* No firm orders were added on #orderdate
then jump EXIT
SEQ AC REQ
-
++ FORMAT1
```

```

DEFINE ELE GREQ AS $CAP(@REQ)
SET FORMAT $REPORT ac(1,6,heading='') greq(+3,10,heading='')
SET FORMAT * grouped by ac (nofinal) greq (nofinal)
SET FORMAT * define value price.sum as sum price
SET FORMAT * define value title.count as count title
SET FORMAT * at end of group greq, in row 1, print(20,8) '$'@price.sum
SET FORMAT * print (30) 'for '@title.count ' title(s)'
SET FORMAT * at end of group ac, in row 1, print(1) @ac ': '$@price.sum
SET FORMAT * print (15) 'for '@title.count ' title(s)'
SET FORMAT * in row +1, print (left) '-----'
SET FORMAT * at start of page, print // //$date ' Page ' $pageno
SET FORMAT * in row +1, print 'Price report for FIRM orders added ' @@DA
SET FORMAT * at end of report, print (left) '_____'
/SET FORMAT * in row +2, print(1) 'Summary: FIRM orders for #orderdate'
SET FORMAT * in row +2, print (1) 'Number of titles: ' @title.count
SET FORMAT * in row +1, print (1) ' Total cost: '$ @price.sum
-
++ TYPE1
IN ACTIVE TYPE
-
++ PRINTOUT
SETP 3800L8
PRINT OUTPUT FILE (cc
++ EXIT
SET ACTIVE ACTIVE FILE
SETP RESTORE

```

APPENDIX K. SEARCHER PROTOCOL

This protocol sets a format found useful for the purposes of the search staff at Virginia Tech.

```
- ! set no echo
SET FORMAT $report TITLE(1,,INDENT=-2,HEADING="")
SET FORMAT * IN ROW +1 AUTHOR(+5,40,wrap,HEADING="")
SET FORMAT * + PUBLISHER(+1,10,HEADING="") YEAR(+2,4,HEADING="")
SET FORMAT * + PLACE(+1,15,HEADING="")
set format * in row +1 instructions(+10,40,heading="")
SET FORMAT * + REQUESTOR(+1,10,HEADING="") AC(+1,4,HEADING="")
SET FORMAT * in row +1 ID(+25,5,HEADING="")
SET FORMAT * + DA(+2,8,HEADING="")
SET FORMAT * + VENDOR(+2,15,HEADING="") announced(+2,10,heading="")
SET FORMAT * + available(+2,10,heading="")
SET FORMAT * SKIP 1
SET FORMAT * AT START OF PAGE, PRINT ""//""//""
clear elements
set active active file
```

APPENDIX L. STPROTOS FILE DEFINER

STPROTOS is a public subfile which contains the protocols shared by all users of the database, both public and private. This includes the protocols which generate the opening messages when the database is selected and the protocol which generates several standard reports for users in the private mode. The file definition was generated from this file definer by the SPIRES FILE DEFINER subsystem.

```
FILE STPROTOS/ AUTHOR H. M. KRIZ, SCI-TECH DEPARTMENT, NEWMAN LIBRARY, 6354
GOAL/ RESULT PROTO
REQUIRED
ELEMENT PROTO.NAME, PROTO/ KEY/ CAPITALIZE
OPTIONAL
ELEMENT COMMAND, COM, C/ XEQ
SUBFILE STPROTOS/ ACCOUNTS SCIBASE
SUBFILE STPROTOS/ ACCOUNTS PUBLIC/SWITCHES 3
  CMD CLEAR SELECT
END
```

APPENDIX M. STPROTOS FILE DEFINITION

STPROTOS is a public subfile which contains the protocols shared by all users of the database, both public and private. This includes the protocols which generate the opening messages when the database is selected and the protocol which generates several standard reports for users in the private mode.

```
FILE = SCIBASE : STPROTOS;
COMMENTS = Generated by FILE DEFINER 06/06/85;
AUTHOR = H. M. KRIZ, SCI-TECH DEPARTMENT, NEWMAN LIBRARY, 6354;
RECORD-NAME = REC01;
  REMOVED;
  REQUIRED;
    KEY = PROTO.NAME;
    ALIASES = PROTO;
    OCC = 1;
    INPROC = $CAP;
  OPTIONAL;
    ELEM = COMMAND;
    ALIASES = COM, C;
    TYPE = XEQ;
SUBFILE-NAME = STPROTOS;
GOAL-RECORD = REC01;
  ACCOUNTS = SCIBASE;
GOAL-RECORD = REC01;
  ACCOUNTS = PUBLIC;
  SECURE-SWITCHES = 3;
  SELECT-COMMAND = CLEAR SELECT;
```

APPENDIX N. PUBLIC ENTRY PROTOCOL

This protocol produces the opening messages for users accessing SCIENCE ORDERS as a public database. It is entered as a record in the public subfile STPROTOS and is executed automatically whenever SCIENCE ORDERS is selected by a public mode user. Any user in any mode of access can view the effect of the protocol by entering the command ..ordersentrypub after selecting SCIENCE ORDERS.

! SET NO ECHO

CLEAR

*

*

*

*	SSSSSSSS	SSSSSSSSSSSSSSSS	SSSSSSSSSS
*	ST ST	S	ST ST
*	ST ST	S	ST ST
*	ST	S	ST ST
*	ST	S	ST ST
*	ST	S	ST ST
*	SSSSSSSS	S	ST ST
*	ST	S	ST ST
*	ST	S	ST ST
*	ST	S	ST ST
*	ST ST	S	ST ST
*	ST ST	S	ST ST
*	SSSSSSSS	S	SSSSSSSSSS

*

*

*

* (Press <RETURN> to continue)

PAUSE

CLEAR

*

*

*

*	+++++		+++++
*	+		+
*	+	SCIENCE ORDERS	+
*	+		+
*	+	A SPIRES software application of the	+
*	+		+
*	+	SCIENCE & TECHNOLOGY DEPARTMENT	+
*	+		+
*	+	Carol M. Newman Library	+
*	+	Virginia Tech	+
*	+	Blacksburg, Virginia	+

```

*      +
*      +-----+
*      +-----+
*      +-----+
*      +-----+
*      +-----+
* (Press <RETURN> to continue)
PAUSE
CLEAR
*
LET T = $RETYPE($UTIME, INT)*2/3600
IF #T<18: IF #T>12: *
                                GOOD AFTERNOON
IF #T=12: *
                                GOOD AFTERNOON
IF #T<12: *
                                GOOD MORNING
IF #T=18: *
                                GOOD EVENING
IF #T>18: *
                                GOOD EVENING
SET ACTIVE ACTIVE FILE
*
*      This is the online database SCIENCE ORDERS operating at the
*      SPIRES command level.
*
/*      SCIENCE ORDERS contains records for $SUBFSIZE titles ordered by
*      the Science and Technology Department, including all Yankee and
*      Taylor-Carlisle approval titles scheduled for shipment during the
*      next several months.
*
*      This file is purged periodically, but only after received items
*      are announced in the Department's list of Selected New Books.
*
*      For the most recent status of items on order, consult a librarian
*      or the Acquisitions Department's In-Process-List.
*
*      Search this file using standard SPIRES procedures.
*      Your active file is set to ACTIVE FILE, the format is $PROMPT.

```

APPENDIX O. PRIVATE ENTRY PROTOCOL

This protocol produces the opening messages for users accessing SCIENCE ORDERS as a private database. It is entered as a record in the public subfile STPROTOS and is executed automatically whenever SCIENCE ORDERS is selected by a private mode user. Any user in any mode of access can view the effect of the protocol by entering the command ..ordersentrypriv after selecting SCIENCE ORDERS.

```
!SET NO ECHO
CLEAR
LET T = $RETYPE($UTIME, INT)*2/3600
IF #T<18: IF #T>12: *                                GOOD AFTERNOON
IF #T=12: *                                GOOD AFTERNOON
IF #T<12: *                                GOOD MORNING
IF #T=18: *                                GOOD EVENING
IF #T>18: *                                GOOD EVENING
SET ACTIVE ACTIVE FILE
*
*      This is the online database SCIENCE ORDERS operating at the
*      SPIRES command level.
*
/*      SCIENCE ORDERS contains records for $SUBFSIZE titles ordered by
*      the Science and Technology Department, including all Yankee and
*      Taylor-Carlisle approval titles scheduled for shipment during the
*      next several months.
*
*      This file is purged periodically, but only after received items
*      are announced in the Department's list of Selected New Books.
*
*      For the most recent status of items on order, consult a librarian
*      or the Acquisitions Department's In-Process-List.
*
*      Search this file using standard SPIRES procedures.
*      Your active file is set to ACTIVE FILE, the format is $PROMPT.
*
*      To generate a formatted printout for any search result
*      enter the command  ..PRINTOUT
```

APPENDIX P. PRINTIPL SPISORT

This protocol prints a listing of all books on order. The protocol has not been used since the end of 1985, since the paper list was found to have no use once the database could be accessed from any userid. If such a complete printout is desired, the protocol should be reviewed for correctness as some procedures may be out of date. Note that for economy the titles are sorted only on the first 20 characters. This is easily changed by editing the L = # of characters in line 12 of the protocol. Care should be taken that sufficient disk space exists to contain the OUTPUT FILE. The file PRINTIPL SPISORT containing this protocol must be on the userid. The procedure is invoked by issuing the commands:

```
USE PRINTIPL SPISORT
XEQ
```

```
! set no echo
CLEAR
* Beginning IPL protocol to generate list of sci-tech books on order.
*
* Erasing old sort files
erase titles unsorted a
erase titles sorted a
erase ipl file
* Generating set of unsorted titles
ldisk syncsort
for subfile
define set titles unsorted a, elem title(L=20)
generate set
endfor
* Sorting titles
spisort titles unsorted a titles sorted a
CLEAR
* Title sort completed.
setp 3800L10
* Printer tag set.
*
* Now setting format, patience please.
SET FORMAT $REPORT id(1,5) da(+1,8,heading='Date//ordered)
SET FORMAT * + Title(+3,50,indent=-2) Author(+1,20,indent=-2)
SET FORMAT * + Publisher(+1,20,trunc) Year(+2,4,trunc)
SET FORMAT * + Requestor(+1,10,indent=-1) ac(+1,4,heading='ACNT')
SET FORMAT * IN ROW +1 vend(+27,15,heading='',trunc)
    available(+2,8,heading='',trunc)
SET FORMAT * skip 1
SET FORMAT * AT START OF PAGE,
    print //'SCI-TECH ORDERS LIST'//$date '      page ' $pageno
SET FORMAT * define value title.count as count title
```

```

SET FORMAT * AT END OF REPORT, print(left) '-----'
SET FORMAT * in row 4, print 'Number of titles: ' @title.count
SET FORMAT * OPTIONS lines = 72
* Format set for IPL
set active ipl file
* Active file set to ipl file
* Writing orders to the active file (Patience please, takes a few minutes)
for set titles sorted a
in active clear display all
* Sending list to the printer.
PRINT IPL FILE (CC
set active active file
setp 3800p6 pre
endfor
*
*
* IPL has been sent to the printer.
*
* Active file has been reset to ACTIVE FILE.
* Printer reset to Prestige 12 pitch
*
* RETURNING TO SPIRES COMMAND MODE.

```